INDIAN INSTITUTE OF TECHNOLOGY, KHARAGPUR

AI60006: Dependable & Secure AI-ML

Assignment 1: Adversarial ML

Gajula Sai Chaitanya **18CS30018**

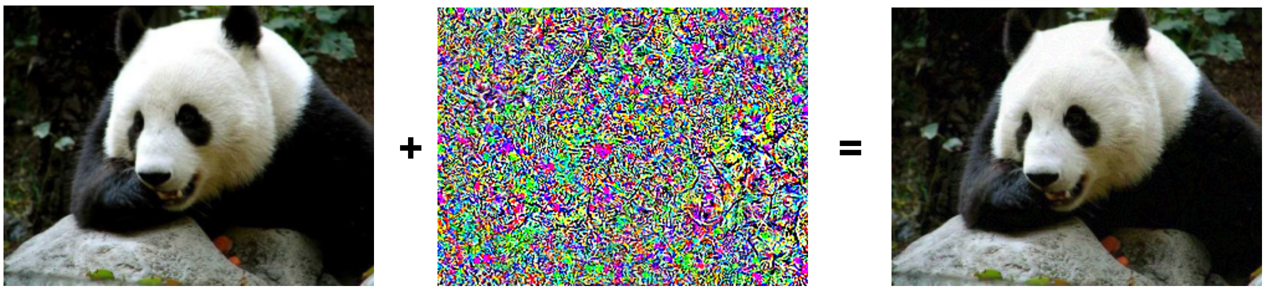
I. Introduction

Adversarial machine learning is a machine learning paradigm that aims to trick machine learning models by providing deceptive input. Hence, it includes both the generation and detection of adversarial examples, which are inputs specially created to deceive classifiers. An Adversarial Attack is a technique to find a perturbation that changes the prediction of a machine learning model. The perturbation can be very small and imperceptible to human eyes.

II. Creating an Adversarial example

Use pretrained Resnet-50 to classify the image

- Take an image and find its class
  - Chose a sample example imagenet from "**giant_panda**" class (Image Source)
- Create an adversarial example for a targeted attack by adding random noise to the image
  - Took **epsilon** value of **4/255**
- Feed it to the Resenet for prediction and find the predicted class of the image



Giant_panda (original_label) **noise** **American_Staffordshire_terrier (misclassified by resnet-50)**

CODE FOR PART 1:
https://www.kaggle.com/code/gsaichaitanya/part1-dsaiml-a1

III. Adversarial training:

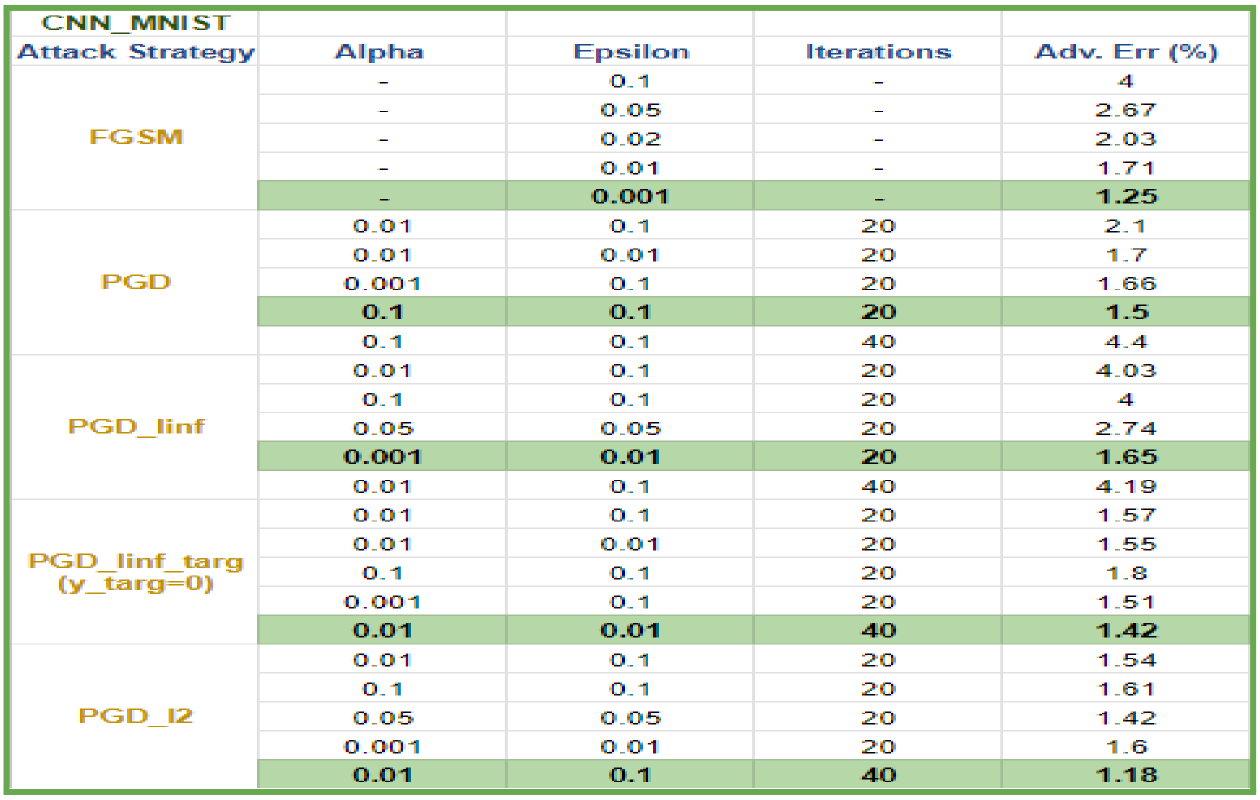
- Constructing adversarial examples using,
    - Fast Gradient Sign Method (FGSM) by varying epsilon between [0.1, 0.001](randomly take five values)
    - Projected gradient descent (PGD)
    - PGD with infinity norm
    - PGD infinity norm with a targeted attack
    - PGD l2 norm by varying epsilon, alpha between [0.1, 0.001] and performing adversarial training to compare the misclassification rate relative to all methods.

- Datasets used:
    - CIFAR10
    - MNIST

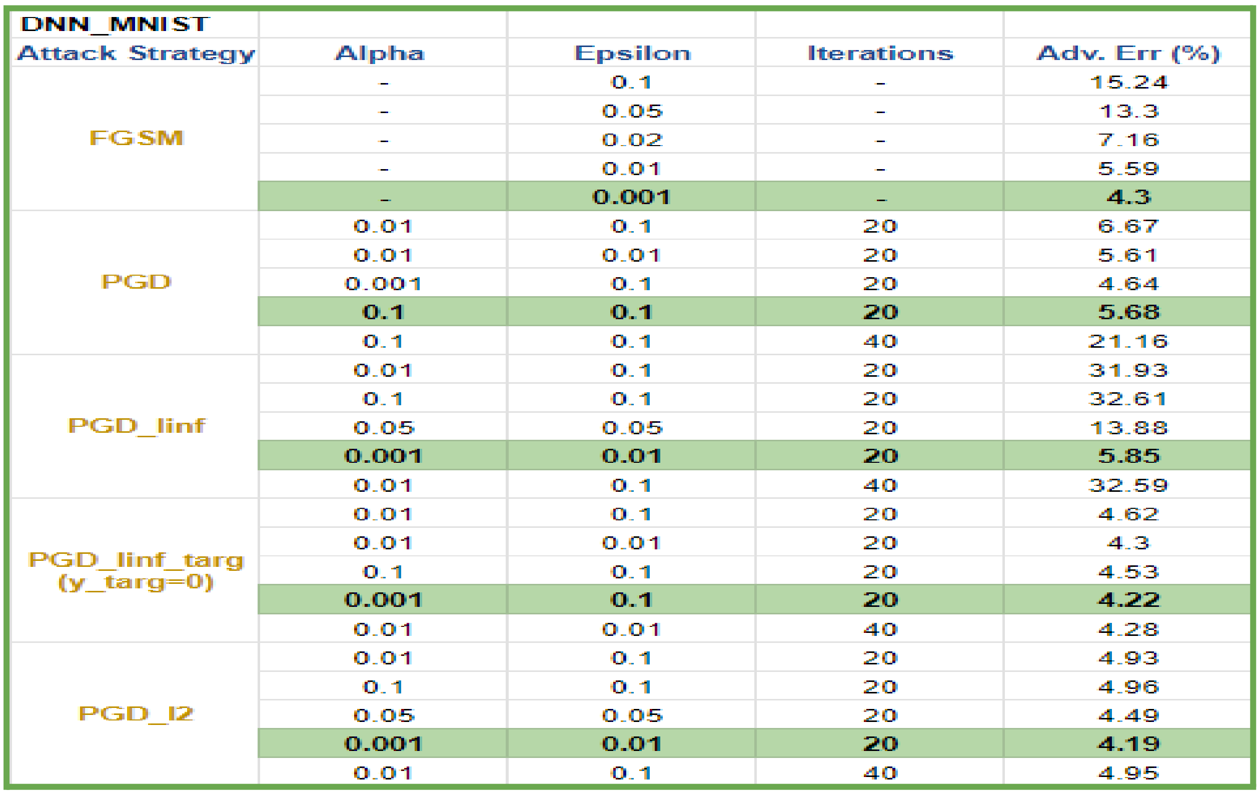
- Models used are:
    - a 6-layer CNN and
    - a 2-layer MLP (DNN)

- The results can be found here:
  DSAIML A1 - 18CS30018

Adversarial Misclassification Rates :



CNN_MNIST				
Attack Strategy	Alpha	Epsilon	Iterations	Adv. Err (%)
FGSM	-	0.1	-	4
	-	0.05	-	2.67
	-	0.02	-	2.03
	-	0.01	-	1.71
	-	**0.001**	**-**	**1.25**
PGD	0.01	0.1	20	2.1
	0.01	0.01	20	1.7
	0.001	0.1	20	1.66
	0.1	**0.1**	**20**	**1.5**
	0.1	0.1	40	4.4
PGD_linf	0.01	0.1	20	4.03
	0.1	0.1	20	4
	0.05	0.05	20	2.74
	0.001	**0.01**	**20**	**1.65**
	0.01	0.1	40	4.19
PGD_linf_targ (y_targ=0)	0.01	0.1	20	1.57
	0.01	0.01	20	1.55
	0.1	0.1	20	1.8
	0.001	0.1	20	1.51
	0.01	**0.01**	**40**	**1.42**
PGD_l2	0.01	0.1	20	1.54
	0.1	0.1	20	1.61
	0.05	0.05	20	1.42
	0.001	0.01	20	1.6
	0.01	**0.1**	**40**	**1.18**



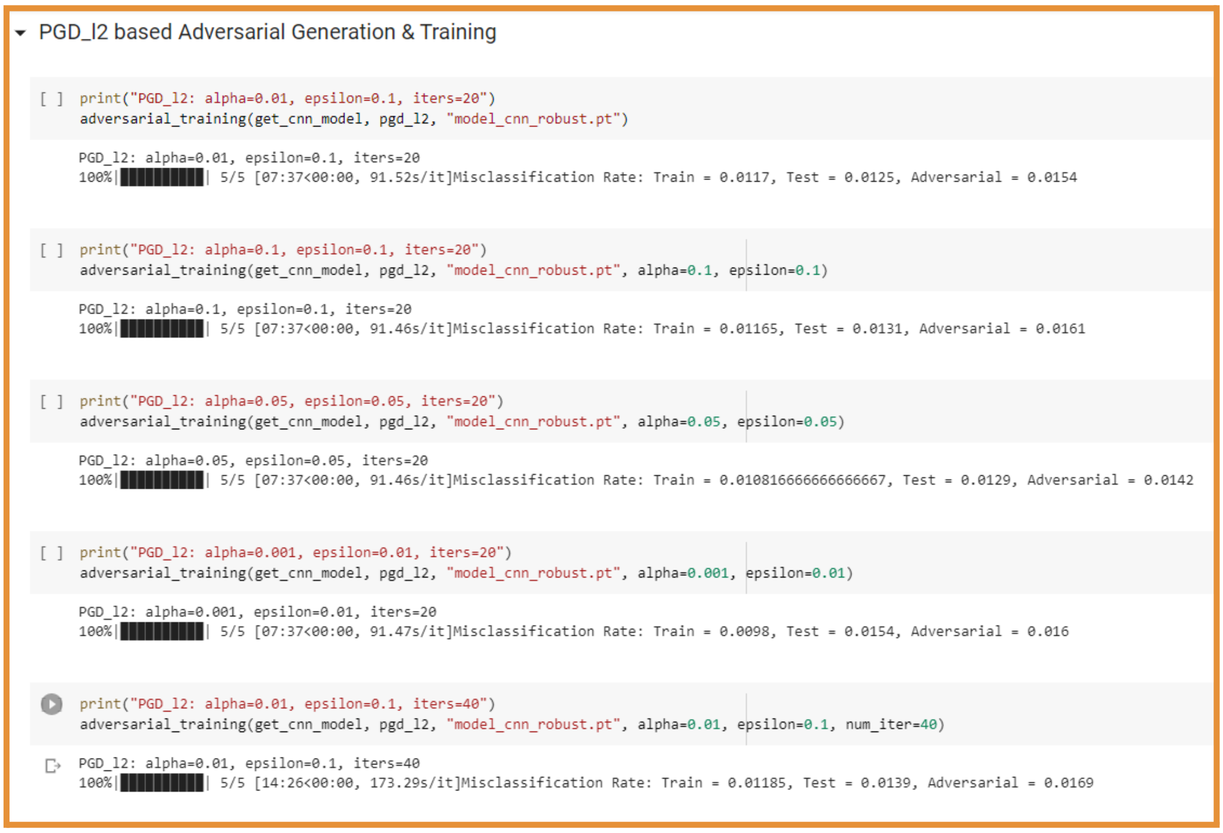
DNN_MNIST				
Attack Strategy	Alpha	Epsilon	Iterations	Adv. Err (%)
FGSM	-	0.1	-	15.24
	-	0.05	-	13.3
	-	0.02	-	7.16
	-	0.01	-	5.59
	-	**0.001**	**-**	**4.3**
PGD	0.01	0.1	20	6.67
	0.01	0.01	20	5.61
	0.001	0.1	20	4.64
	0.1	**0.1**	**20**	**5.68**
	0.1	0.1	40	21.16
PGD_linf	0.01	0.1	20	31.93
	0.1	0.1	20	32.61
	0.05	0.05	20	13.88
	0.001	**0.01**	**20**	**5.85**
	0.01	0.1	40	32.59
PGD_linf_targ (y_targ=0)	0.01	0.1	20	4.62
	0.01	0.01	20	4.3
	0.1	0.1	20	4.53
	0.001	**0.1**	**20**	**4.22**
	0.01	0.01	40	4.28
PGD_l2	0.01	0.1	20	4.93
	0.1	0.1	20	4.96
	0.05	0.05	20	4.49
	0.001	**0.01**	**20**	**4.19**
	0.01	0.1	40	4.95

DNN_CIFAR				
Attack Strategy	Alpha	Epsilon	Iterations	Adv. Err (%)
FGSM	-	0.1	-	90.01
	-	0.05	-	90.03
	-	0.02	-	90.03
	-	0.01	-	75.17
	-	**0.001**	**-**	**59.29**
PGD	0.01	0.1	20	60.6
	0.01	0.01	20	60.95
	0.001	**0.1**	**20**	**57.68**
	0.1	0.1	20	68.61
	0.1	0.1	40	78.42
PGD_linf	0.01	0.1	20	90.02
	0.1	0.1	20	90.01
	0.05	0.05	20	90.01
	0.001	**0.01**	**20**	**68.77**
	0.01	0.1	40	90.02
PGD_linf_targ (y_targ=0)	0.01	0.1	20	59.81
	0.01	0.01	20	56.95
	0.1	0.1	20	60.52
	0.001	0.1	20	58.65
	0.01	**0.01**	**40**	**57.31**
PGD_l2	0.01	0.1	20	59.52
	0.1	0.1	20	60.99
	0.05	0.05	20	59.7
	0.001	0.01	20	57.85
	0.01	0.1	40	60.13

CNN_CIFAR				
Attack Strategy	Alpha	Epsilon	Iterations	Adv. Err (%)
FGSM	-	0.1	-	90
	-	0.05	-	76.54
	-	0.02	-	76.63
	-	0.01	-	57.7
	-	**0.001**	**-**	**44.38**
PGD	0.01	0.1	20	51.27
	0.01	0.01	20	50.76
	0.001	**0.1**	**20**	**44.39**
	0.1	0.1	20	56.88
	0.1	0.1	40	67.48
PGD_linf	0.01	0.1	20	90
	0.1	0.1	20	90.43
	0.05	0.05	20	76.63
	0.001	**0.01**	**20**	**57.83**
	0.01	0.1	40	90
PGD_linf_targ ()y_targ=0)	0.01	0.1	20	51.72
	0.01	0.01	20	43.43
	0.1	0.1	20	49.31
	0.001	0.1	20	43.13
	0.01	**0.01**	**40**	**41.76**
PGD_l2	0.01	0.1	20	47.69
	0.1	0.1	20	49.49
	0.05	0.05	20	44.9
	0.001	**0.01**	**20**	**43.69**
	0.01	0.1	40	48.98

SCREENSHOTS:

1) CNN_MNIST:



PGD_l2 based Adversarial Generation & Training

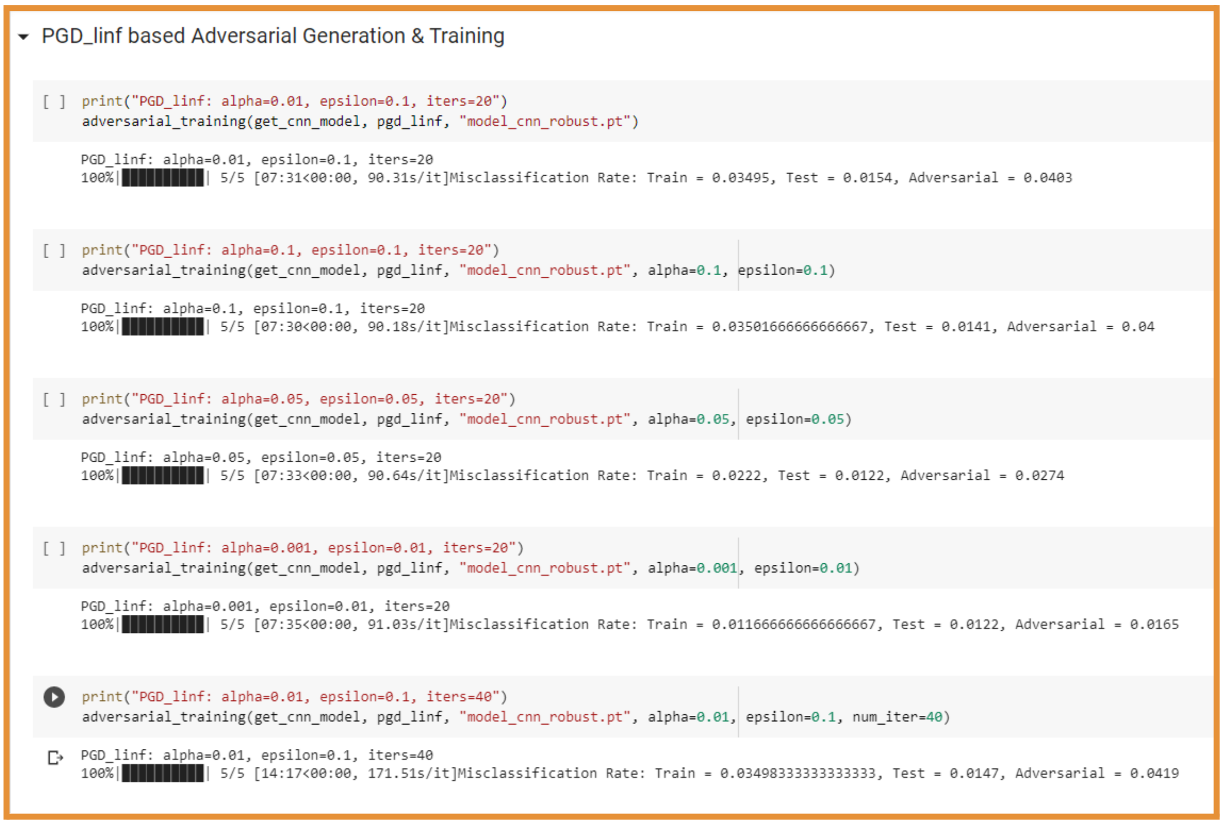
```
print("PGD_l2: alpha=0.01, epsilon=0.1, iters=20")
adversarial_training(get_cnn_model, pgd_l2, "model_cnn_robust.pt")

PGD_l2: alpha=0.01, epsilon=0.1, iters=20
100%|██████████| 5/5 [07:37<00:00, 91.52s/it]Misclassification Rate: Train = 0.0117, Test = 0.0125, Adversarial = 0.0154
```

```
print("PGD_l2: alpha=0.1, epsilon=0.1, iters=20")
adversarial_training(get_cnn_model, pgd_l2, "model_cnn_robust.pt", alpha=0.1, epsilon=0.1)

PGD_l2: alpha=0.1, epsilon=0.1, iters=20
100%|██████████| 5/5 [07:37<00:00, 91.46s/it]Misclassification Rate: Train = 0.01165, Test = 0.0131, Adversarial = 0.0161
```

```
print("PGD_l2: alpha=0.05, epsilon=0.05, iters=20")
adversarial_training(get_cnn_model, pgd_l2, "model_cnn_robust.pt", alpha=0.05, epsilon=0.05)

PGD_l2: alpha=0.05, epsilon=0.05, iters=20
100%|██████████| 5/5 [07:37<00:00, 91.46s/it]Misclassification Rate: Train = 0.010816666666666667, Test = 0.0129, Adversarial = 0.0142
```

```
print("PGD_l2: alpha=0.001, epsilon=0.01, iters=20")
adversarial_training(get_cnn_model, pgd_l2, "model_cnn_robust.pt", alpha=0.001, epsilon=0.01)

PGD_l2: alpha=0.001, epsilon=0.01, iters=20
100%|██████████| 5/5 [07:37<00:00, 91.47s/it]Misclassification Rate: Train = 0.0098, Test = 0.0154, Adversarial = 0.016
```

```
print("PGD_l2: alpha=0.01, epsilon=0.1, iters=40")
adversarial_training(get_cnn_model, pgd_l2, "model_cnn_robust.pt", alpha=0.01, epsilon=0.1, num_iter=40)

PGD_l2: alpha=0.01, epsilon=0.1, iters=40
100%|██████████| 5/5 [14:26<00:00, 173.29s/it]Misclassification Rate: Train = 0.01185, Test = 0.0139, Adversarial = 0.0169
```



PGD_linf based Adversarial Generation & Training

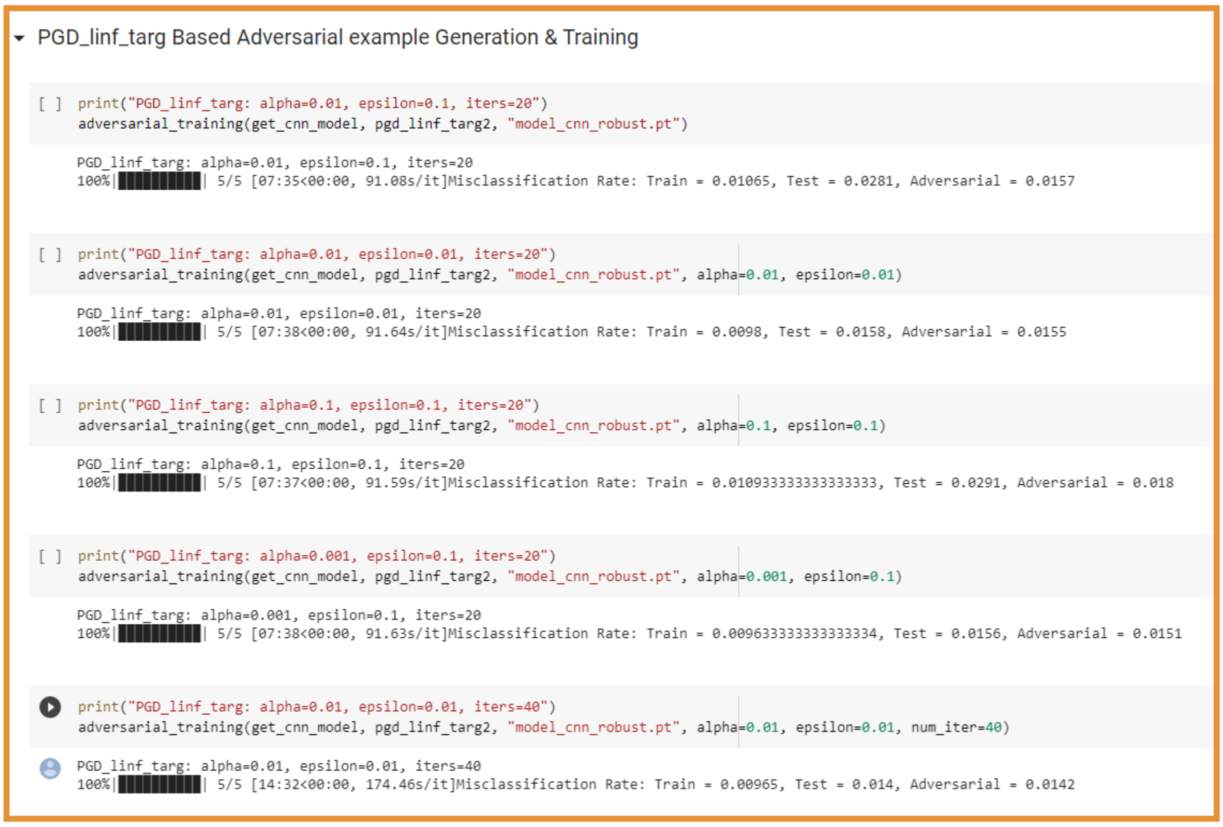
```
print("PGD_linf: alpha=0.01, epsilon=0.1, iters=20")
adversarial_training(get_cnn_model, pgd_linf, "model_cnn_robust.pt")

PGD_linf: alpha=0.01, epsilon=0.1, iters=20
100%|██████████| 5/5 [07:31<00:00, 90.31s/it]Misclassification Rate: Train = 0.03495, Test = 0.0154, Adversarial = 0.0403
```

```
print("PGD_linf: alpha=0.1, epsilon=0.1, iters=20")
adversarial_training(get_cnn_model, pgd_linf, "model_cnn_robust.pt", alpha=0.1, epsilon=0.1)

PGD_linf: alpha=0.1, epsilon=0.1, iters=20
100%|██████████| 5/5 [07:30<00:00, 90.18s/it]Misclassification Rate: Train = 0.03501666666666667, Test = 0.0141, Adversarial = 0.04
```

```
print("PGD_linf: alpha=0.05, epsilon=0.05, iters=20")
adversarial_training(get_cnn_model, pgd_linf, "model_cnn_robust.pt", alpha=0.05, epsilon=0.05)

PGD_linf: alpha=0.05, epsilon=0.05, iters=20
100%|██████████| 5/5 [07:33<00:00, 90.64s/it]Misclassification Rate: Train = 0.0222, Test = 0.0122, Adversarial = 0.0274
```

```
print("PGD_linf: alpha=0.001, epsilon=0.01, iters=20")
adversarial_training(get_cnn_model, pgd_linf, "model_cnn_robust.pt", alpha=0.001, epsilon=0.01)

PGD_linf: alpha=0.001, epsilon=0.01, iters=20
100%|██████████| 5/5 [07:35<00:00, 91.03s/it]Misclassification Rate: Train = 0.011666666666666667, Test = 0.0122, Adversarial = 0.0165
```

```
print("PGD_linf: alpha=0.01, epsilon=0.1, iters=40")
adversarial_training(get_cnn_model, pgd_linf, "model_cnn_robust.pt", alpha=0.01, epsilon=0.1, num_iter=40)

PGD_linf: alpha=0.01, epsilon=0.1, iters=40
100%|██████████| 5/5 [14:17<00:00, 171.51s/it]Misclassification Rate: Train = 0.03498333333333333, Test = 0.0147, Adversarial = 0.0419
```



PGD_linf_targ Based Adversarial example Generation & Training

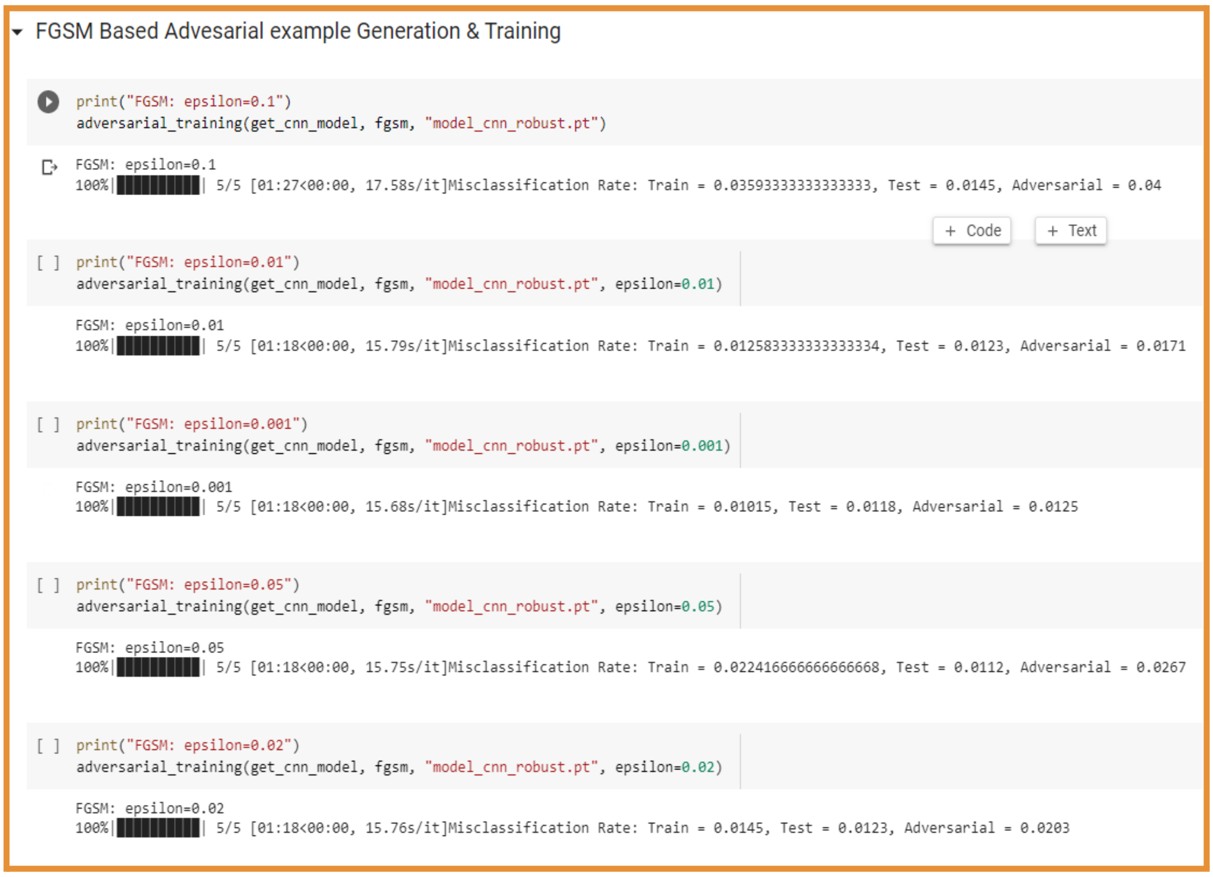
```
[ ] print("PGD_linf_targ: alpha=0.01, epsilon=0.1, iters=20")
    adversarial_training(get_cnn_model, pgd_linf_targ2, "model_cnn_robust.pt")

    PGD_linf_targ: alpha=0.01, epsilon=0.1, iters=20
    100%|██████████| 5/5 [07:35<00:00, 91.08s/it]Misclassification Rate: Train = 0.01065, Test = 0.0281, Adversarial = 0.0157
```

```
[ ] print("PGD_linf_targ: alpha=0.01, epsilon=0.01, iters=20")
    adversarial_training(get_cnn_model, pgd_linf_targ2, "model_cnn_robust.pt", alpha=0.01, epsilon=0.01)

    PGD_linf_targ: alpha=0.01, epsilon=0.01, iters=20
    100%|██████████| 5/5 [07:38<00:00, 91.64s/it]Misclassification Rate: Train = 0.0098, Test = 0.0158, Adversarial = 0.0155
```

```
[ ] print("PGD_linf_targ: alpha=0.1, epsilon=0.1, iters=20")
    adversarial_training(get_cnn_model, pgd_linf_targ2, "model_cnn_robust.pt", alpha=0.1, epsilon=0.1)

    PGD_linf_targ: alpha=0.1, epsilon=0.1, iters=20
    100%|██████████| 5/5 [07:37<00:00, 91.59s/it]Misclassification Rate: Train = 0.010933333333333333, Test = 0.0291, Adversarial = 0.018
```

```
[ ] print("PGD_linf_targ: alpha=0.001, epsilon=0.1, iters=20")
    adversarial_training(get_cnn_model, pgd_linf_targ2, "model_cnn_robust.pt", alpha=0.001, epsilon=0.1)

    PGD_linf_targ: alpha=0.001, epsilon=0.1, iters=20
    100%|██████████| 5/5 [07:38<00:00, 91.63s/it]Misclassification Rate: Train = 0.009633333333333334, Test = 0.0156, Adversarial = 0.0151
```

```
print("PGD_linf_targ: alpha=0.01, epsilon=0.01, iters=40")
adversarial_training(get_cnn_model, pgd_linf_targ2, "model_cnn_robust.pt", alpha=0.01, epsilon=0.01, num_iter=40)

PGD_linf_targ: alpha=0.01, epsilon=0.01, iters=40
100%|██████████| 5/5 [14:32<00:00, 174.46s/it]Misclassification Rate: Train = 0.00965, Test = 0.014, Adversarial = 0.0142
```



FGSM Based Advesarial example Generation & Training

```
print("FGSM: epsilon=0.1")
adversarial_training(get_cnn_model, fgsm, "model_cnn_robust.pt")
```

```
FGSM: epsilon=0.1
100%|██████████| 5/5 [01:27<00:00, 17.58s/it]Misclassification Rate: Train = 0.035933333333333333, Test = 0.0145, Adversarial = 0.04
```

```
[ ] print("FGSM: epsilon=0.01")
    adversarial_training(get_cnn_model, fgsm, "model_cnn_robust.pt", epsilon=0.01)
```

```
FGSM: epsilon=0.01
100%|██████████| 5/5 [01:18<00:00, 15.79s/it]Misclassification Rate: Train = 0.012583333333333334, Test = 0.0123, Adversarial = 0.0171
```

```
[ ] print("FGSM: epsilon=0.001")
    adversarial_training(get_cnn_model, fgsm, "model_cnn_robust.pt", epsilon=0.001)
```

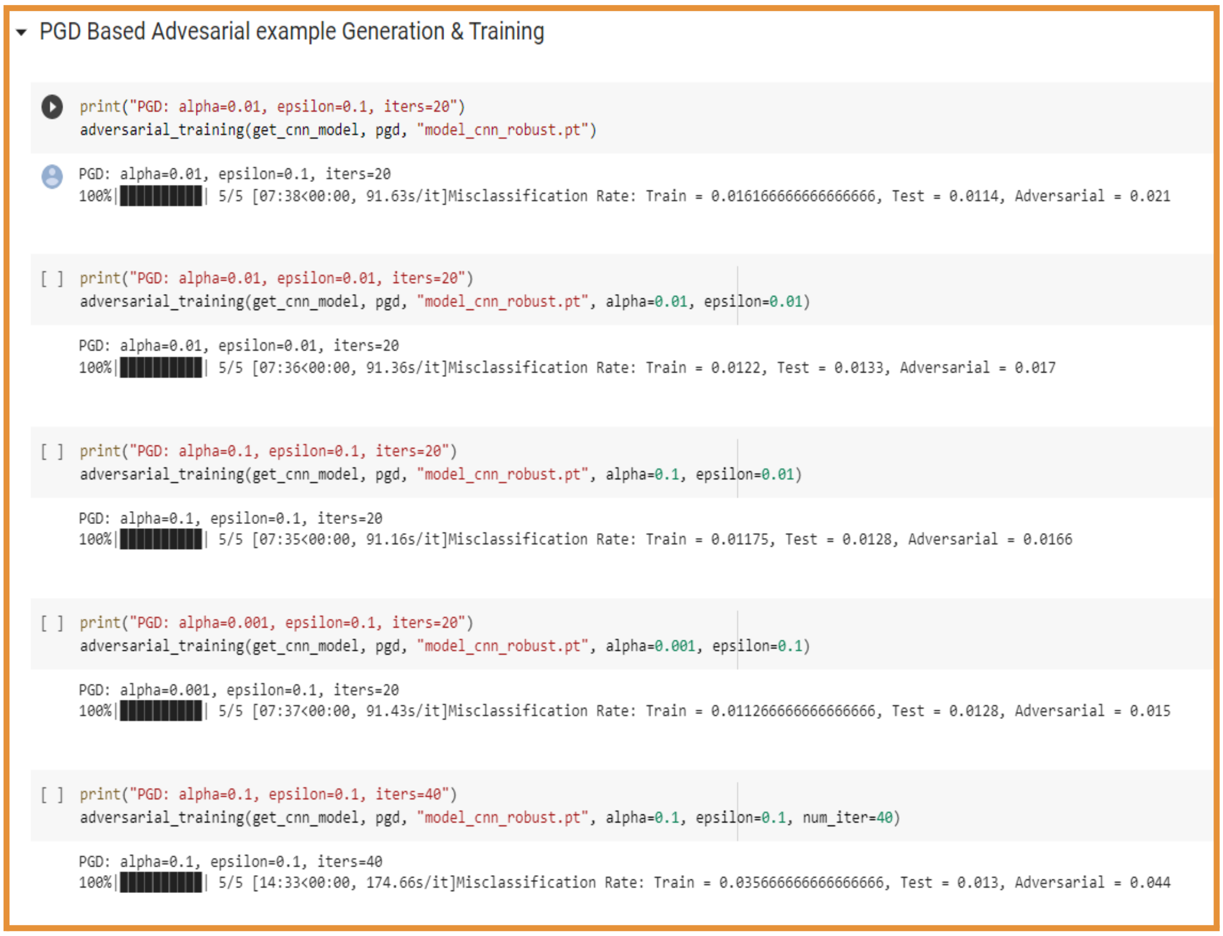
```
FGSM: epsilon=0.001
100%|██████████| 5/5 [01:18<00:00, 15.68s/it]Misclassification Rate: Train = 0.01015, Test = 0.0118, Adversarial = 0.0125
```

```
[ ] print("FGSM: epsilon=0.05")
    adversarial_training(get_cnn_model, fgsm, "model_cnn_robust.pt", epsilon=0.05)
```

```
FGSM: epsilon=0.05
100%|██████████| 5/5 [01:18<00:00, 15.75s/it]Misclassification Rate: Train = 0.022416666666666668, Test = 0.0112, Adversarial = 0.0267
```

```
[ ] print("FGSM: epsilon=0.02")
    adversarial_training(get_cnn_model, fgsm, "model_cnn_robust.pt", epsilon=0.02)
```

```
FGSM: epsilon=0.02
100%|██████████| 5/5 [01:18<00:00, 15.76s/it]Misclassification Rate: Train = 0.0145, Test = 0.0123, Adversarial = 0.0203
```



PGD Based Advesarial example Generation & Training

```
print("PGD: alpha=0.01, epsilon=0.1, iters=20")
adversarial_training(get_cnn_model, pgd, "model_cnn_robust.pt")
```

```
PGD: alpha=0.01, epsilon=0.1, iters=20
100%|██████████| 5/5 [07:38<00:00, 91.63s/it]Misclassification Rate: Train = 0.016166666666666666, Test = 0.0114, Adversarial = 0.021
```

```
[ ] print("PGD: alpha=0.01, epsilon=0.01, iters=20")
    adversarial_training(get_cnn_model, pgd, "model_cnn_robust.pt", alpha=0.01, epsilon=0.01)
```

```
PGD: alpha=0.01, epsilon=0.01, iters=20
100%|██████████| 5/5 [07:36<00:00, 91.36s/it]Misclassification Rate: Train = 0.0122, Test = 0.0133, Adversarial = 0.017
```

```
[ ] print("PGD: alpha=0.1, epsilon=0.1, iters=20")
    adversarial_training(get_cnn_model, pgd, "model_cnn_robust.pt", alpha=0.1, epsilon=0.01)
```

```
PGD: alpha=0.1, epsilon=0.1, iters=20
100%|██████████| 5/5 [07:35<00:00, 91.16s/it]Misclassification Rate: Train = 0.01175, Test = 0.0128, Adversarial = 0.0166
```

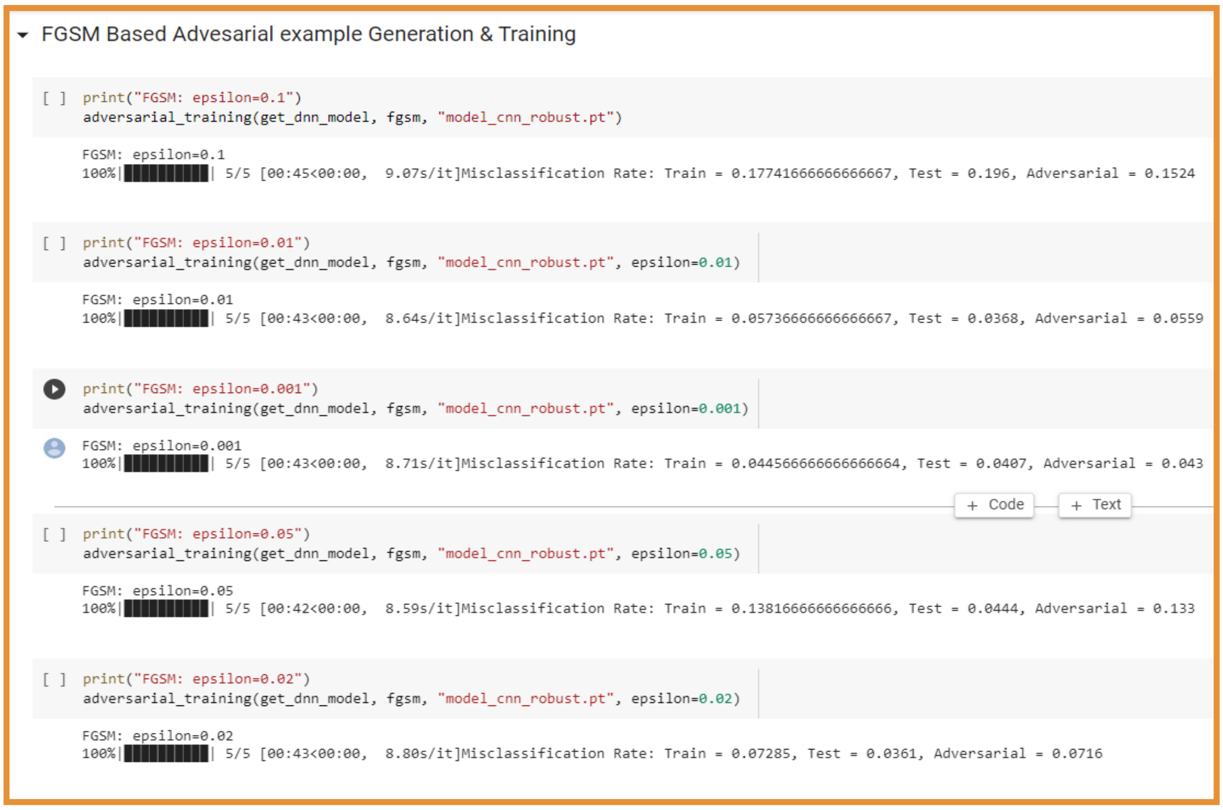
```
[ ] print("PGD: alpha=0.001, epsilon=0.1, iters=20")
    adversarial_training(get_cnn_model, pgd, "model_cnn_robust.pt", alpha=0.001, epsilon=0.1)
```

```
PGD: alpha=0.001, epsilon=0.1, iters=20
100%|██████████| 5/5 [07:37<00:00, 91.43s/it]Misclassification Rate: Train = 0.011266666666666666, Test = 0.0128, Adversarial = 0.015
```

```
[ ] print("PGD: alpha=0.1, epsilon=0.1, iters=40")
    adversarial_training(get_cnn_model, pgd, "model_cnn_robust.pt", alpha=0.1, epsilon=0.1, num_iter=40)
```

```
PGD: alpha=0.1, epsilon=0.1, iters=40
100%|██████████| 5/5 [14:33<00:00, 174.66s/it]Misclassification Rate: Train = 0.035666666666666666, Test = 0.013, Adversarial = 0.044
```

2) DNN_MNIST:



FGSM Based Advesarial example Generation & Training

```
print("FGSM: epsilon=0.1")
adversarial_training(get_dnn_model, fgsm, "model_cnn_robust.pt")
```

```
FGSM: epsilon=0.1
100%|██████████| 5/5 [00:45<00:00,  9.07s/it]Misclassification Rate: Train = 0.17741666666666667, Test = 0.196, Adversarial = 0.1524
```

```
print("FGSM: epsilon=0.01")
adversarial_training(get_dnn_model, fgsm, "model_cnn_robust.pt", epsilon=0.01)
```

```
FGSM: epsilon=0.01
100%|██████████| 5/5 [00:43<00:00,  8.64s/it]Misclassification Rate: Train = 0.05736666666666667, Test = 0.0368, Adversarial = 0.0559
```

```
print("FGSM: epsilon=0.001")
adversarial_training(get_dnn_model, fgsm, "model_cnn_robust.pt", epsilon=0.001)
```

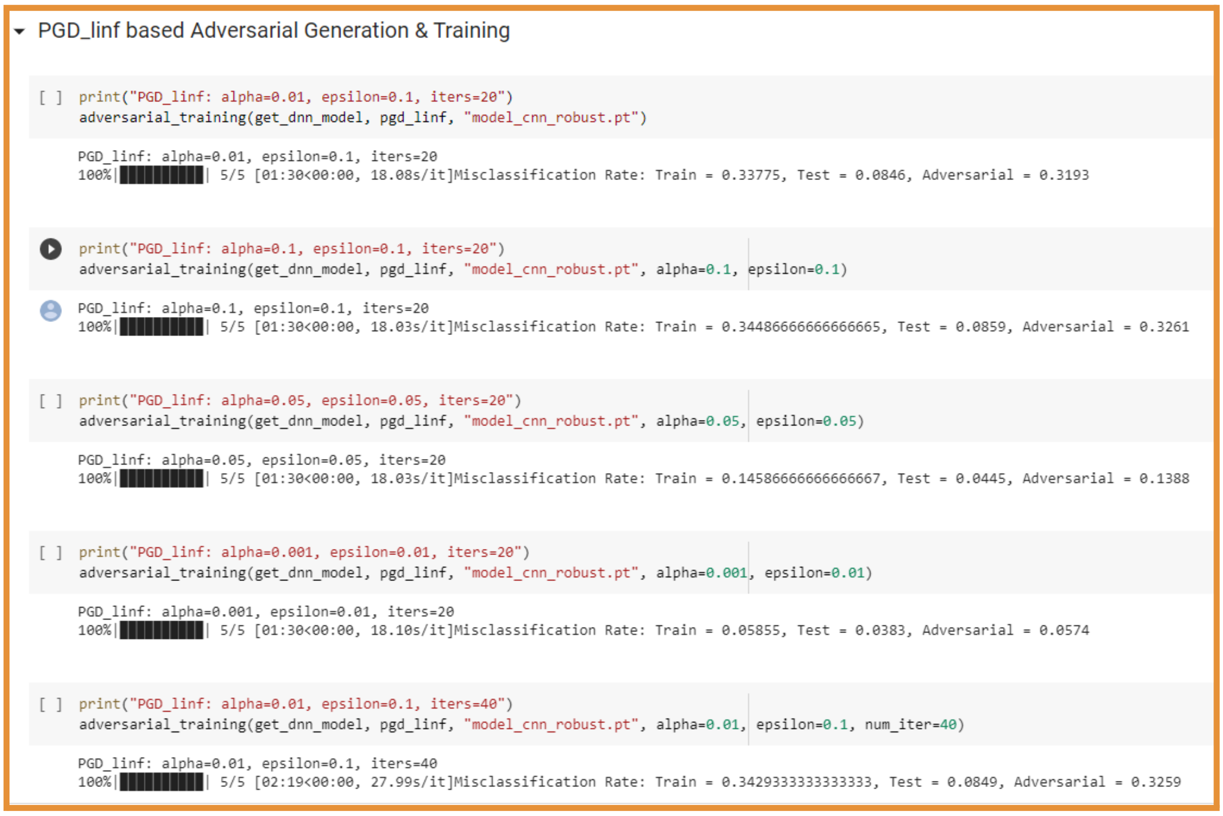
```
FGSM: epsilon=0.001
100%|██████████| 5/5 [00:43<00:00,  8.71s/it]Misclassification Rate: Train = 0.044566666666666664, Test = 0.0407, Adversarial = 0.043
```

```
print("FGSM: epsilon=0.05")
adversarial_training(get_dnn_model, fgsm, "model_cnn_robust.pt", epsilon=0.05)
```

```
FGSM: epsilon=0.05
100%|██████████| 5/5 [00:42<00:00,  8.59s/it]Misclassification Rate: Train = 0.13816666666666666, Test = 0.0444, Adversarial = 0.133
```

```
print("FGSM: epsilon=0.02")
adversarial_training(get_dnn_model, fgsm, "model_cnn_robust.pt", epsilon=0.02)
```

```
FGSM: epsilon=0.02
100%|██████████| 5/5 [00:43<00:00,  8.80s/it]Misclassification Rate: Train = 0.07285, Test = 0.0361, Adversarial = 0.0716
```



PGD_linf based Adversarial Generation & Training

```
print("PGD_linf: alpha=0.01, epsilon=0.1, iters=20")
adversarial_training(get_dnn_model, pgd_linf, "model_cnn_robust.pt")
```

```
PGD_linf: alpha=0.01, epsilon=0.1, iters=20
100%|██████████| 5/5 [01:30<00:00, 18.08s/it]Misclassification Rate: Train = 0.33775, Test = 0.0846, Adversarial = 0.3193
```

```
print("PGD_linf: alpha=0.1, epsilon=0.1, iters=20")
adversarial_training(get_dnn_model, pgd_linf, "model_cnn_robust.pt", alpha=0.1, epsilon=0.1)
```

```
PGD_linf: alpha=0.1, epsilon=0.1, iters=20
100%|██████████| 5/5 [01:30<00:00, 18.03s/it]Misclassification Rate: Train = 0.34486666666666665, Test = 0.0859, Adversarial = 0.3261
```

```
print("PGD_linf: alpha=0.05, epsilon=0.05, iters=20")
adversarial_training(get_dnn_model, pgd_linf, "model_cnn_robust.pt", alpha=0.05, epsilon=0.05)
```

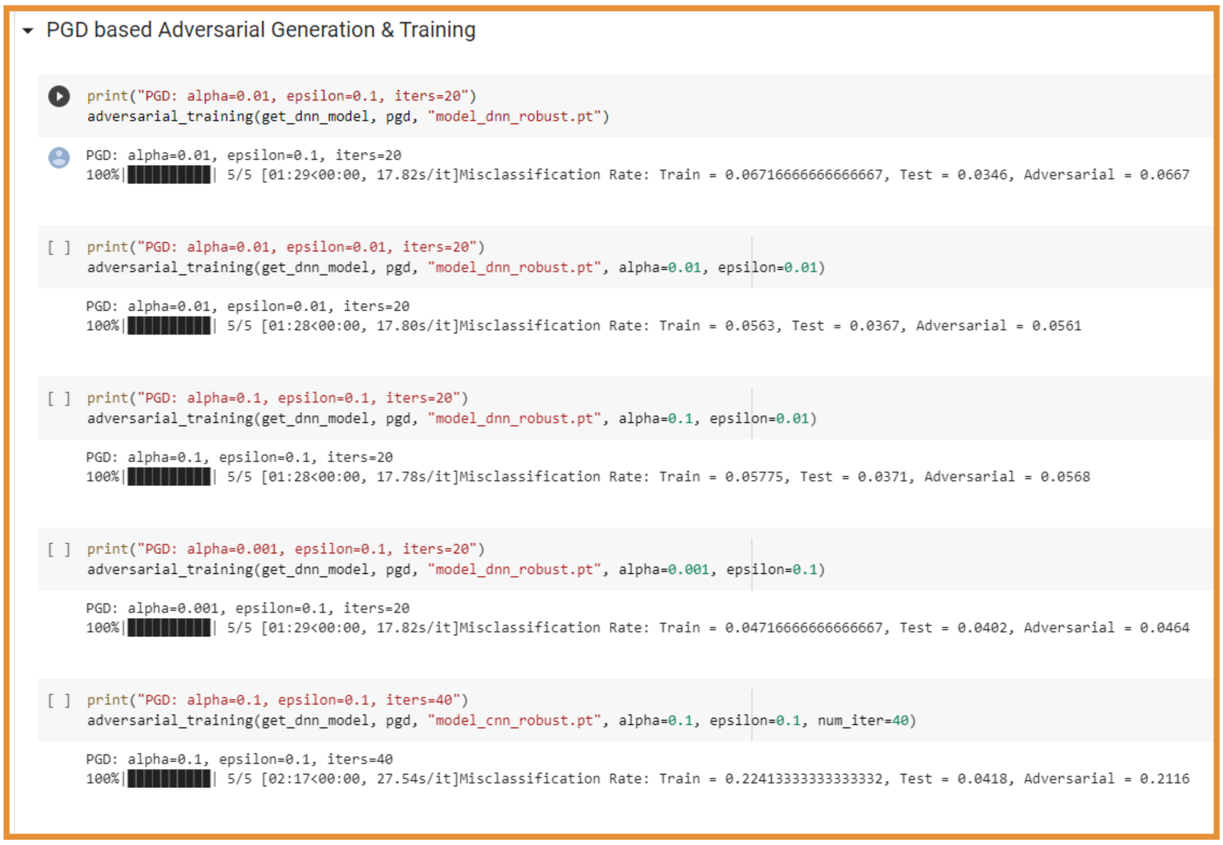
```
PGD_linf: alpha=0.05, epsilon=0.05, iters=20
100%|██████████| 5/5 [01:30<00:00, 18.03s/it]Misclassification Rate: Train = 0.14586666666666667, Test = 0.0445, Adversarial = 0.1388
```

```
print("PGD_linf: alpha=0.001, epsilon=0.01, iters=20")
adversarial_training(get_dnn_model, pgd_linf, "model_cnn_robust.pt", alpha=0.001, epsilon=0.01)
```

```
PGD_linf: alpha=0.001, epsilon=0.01, iters=20
100%|██████████| 5/5 [01:30<00:00, 18.10s/it]Misclassification Rate: Train = 0.05855, Test = 0.0383, Adversarial = 0.0574
```

```
print("PGD_linf: alpha=0.01, epsilon=0.1, iters=40")
adversarial_training(get_dnn_model, pgd_linf, "model_cnn_robust.pt", alpha=0.01, epsilon=0.1, num_iter=40)
```

```
PGD_linf: alpha=0.01, epsilon=0.1, iters=40
100%|██████████| 5/5 [02:19<00:00, 27.99s/it]Misclassification Rate: Train = 0.3429333333333333, Test = 0.0849, Adversarial = 0.3259
```



PGD based Adversarial Generation & Training

```
print("PGD: alpha=0.01, epsilon=0.1, iters=20")
adversarial_training(get_dnn_model, pgd, "model_dnn_robust.pt")
```

```
PGD: alpha=0.01, epsilon=0.1, iters=20
100%|██████████| 5/5 [01:29<00:00, 17.82s/it]Misclassification Rate: Train = 0.06716666666666667, Test = 0.0346, Adversarial = 0.0667
```

```
[ ] print("PGD: alpha=0.01, epsilon=0.01, iters=20")
    adversarial_training(get_dnn_model, pgd, "model_dnn_robust.pt", alpha=0.01, epsilon=0.01)
```

```
PGD: alpha=0.01, epsilon=0.01, iters=20
100%|██████████| 5/5 [01:28<00:00, 17.80s/it]Misclassification Rate: Train = 0.0563, Test = 0.0367, Adversarial = 0.0561
```

```
[ ] print("PGD: alpha=0.1, epsilon=0.1, iters=20")
    adversarial_training(get_dnn_model, pgd, "model_dnn_robust.pt", alpha=0.1, epsilon=0.01)
```

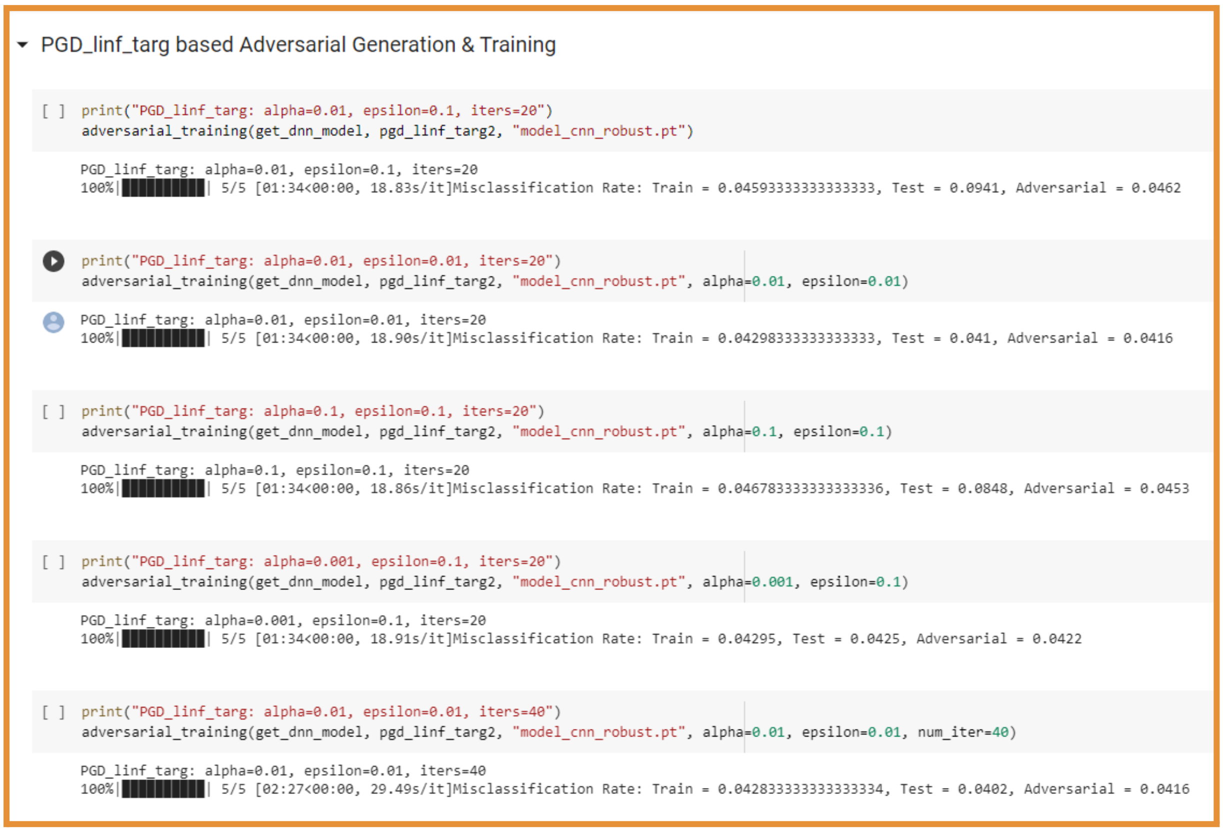
```
PGD: alpha=0.1, epsilon=0.1, iters=20
100%|██████████| 5/5 [01:28<00:00, 17.78s/it]Misclassification Rate: Train = 0.05775, Test = 0.0371, Adversarial = 0.0568
```

```
[ ] print("PGD: alpha=0.001, epsilon=0.1, iters=20")
    adversarial_training(get_dnn_model, pgd, "model_dnn_robust.pt", alpha=0.001, epsilon=0.1)
```

```
PGD: alpha=0.001, epsilon=0.1, iters=20
100%|██████████| 5/5 [01:29<00:00, 17.82s/it]Misclassification Rate: Train = 0.04716666666666667, Test = 0.0402, Adversarial = 0.0464
```

```
[ ] print("PGD: alpha=0.1, epsilon=0.1, iters=40")
    adversarial_training(get_dnn_model, pgd, "model_cnn_robust.pt", alpha=0.1, epsilon=0.1, num_iter=40)
```

```
PGD: alpha=0.1, epsilon=0.1, iters=40
100%|██████████| 5/5 [02:17<00:00, 27.54s/it]Misclassification Rate: Train = 0.22413333333333332, Test = 0.0418, Adversarial = 0.2116
```



PGD_linf_targ based Adversarial Generation & Training

```
[ ] print("PGD_linf_targ: alpha=0.01, epsilon=0.1, iters=20")
    adversarial_training(get_dnn_model, pgd_linf_targ2, "model_cnn_robust.pt")
```

```
PGD_linf_targ: alpha=0.01, epsilon=0.1, iters=20
100%|██████████| 5/5 [01:34<00:00, 18.83s/it]Misclassification Rate: Train = 0.04593333333333333, Test = 0.0941, Adversarial = 0.0462
```

```
print("PGD_linf_targ: alpha=0.01, epsilon=0.01, iters=20")
adversarial_training(get_dnn_model, pgd_linf_targ2, "model_cnn_robust.pt", alpha=0.01, epsilon=0.01)
```

```
PGD_linf_targ: alpha=0.01, epsilon=0.01, iters=20
100%|██████████| 5/5 [01:34<00:00, 18.90s/it]Misclassification Rate: Train = 0.04298333333333333, Test = 0.041, Adversarial = 0.0416
```

```
[ ] print("PGD_linf_targ: alpha=0.1, epsilon=0.1, iters=20")
    adversarial_training(get_dnn_model, pgd_linf_targ2, "model_cnn_robust.pt", alpha=0.1, epsilon=0.1)
```

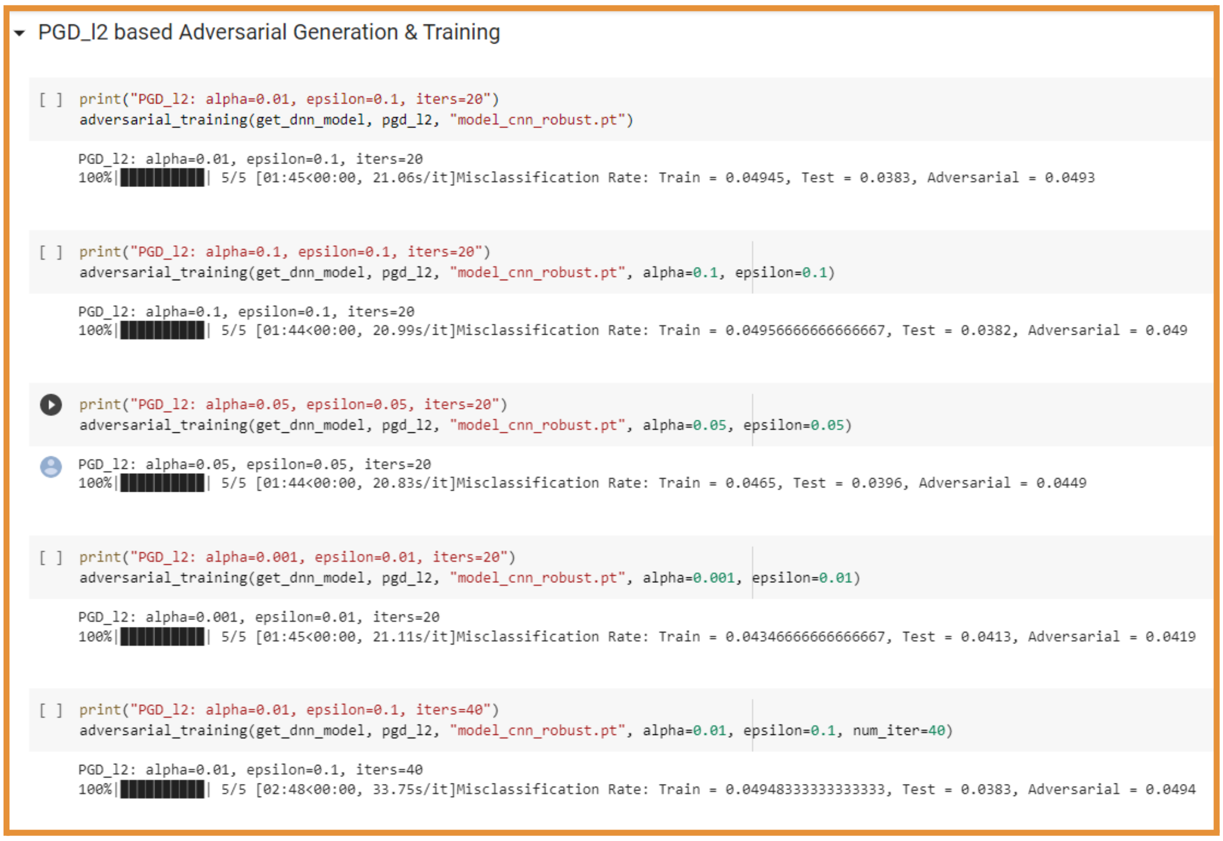
```
PGD_linf_targ: alpha=0.1, epsilon=0.1, iters=20
100%|██████████| 5/5 [01:34<00:00, 18.86s/it]Misclassification Rate: Train = 0.046783333333333336, Test = 0.0848, Adversarial = 0.0453
```

```
[ ] print("PGD_linf_targ: alpha=0.001, epsilon=0.1, iters=20")
    adversarial_training(get_dnn_model, pgd_linf_targ2, "model_cnn_robust.pt", alpha=0.001, epsilon=0.1)
```

```
PGD_linf_targ: alpha=0.001, epsilon=0.1, iters=20
100%|██████████| 5/5 [01:34<00:00, 18.91s/it]Misclassification Rate: Train = 0.04295, Test = 0.0425, Adversarial = 0.0422
```

```
[ ] print("PGD_linf_targ: alpha=0.01, epsilon=0.01, iters=40")
    adversarial_training(get_dnn_model, pgd_linf_targ2, "model_cnn_robust.pt", alpha=0.01, epsilon=0.01, num_iter=40)
```

```
PGD_linf_targ: alpha=0.01, epsilon=0.01, iters=40
100%|██████████| 5/5 [02:27<00:00, 29.49s/it]Misclassification Rate: Train = 0.042833333333333334, Test = 0.0402, Adversarial = 0.0416
```



PGD_l2 based Adversarial Generation & Training

```
[ ] print("PGD_l2: alpha=0.01, epsilon=0.1, iters=20")
    adversarial_training(get_dnn_model, pgd_l2, "model_cnn_robust.pt")

    PGD_l2: alpha=0.01, epsilon=0.1, iters=20
    100%|██████████| 5/5 [01:45<00:00, 21.06s/it]Misclassification Rate: Train = 0.04945, Test = 0.0383, Adversarial = 0.0493
```

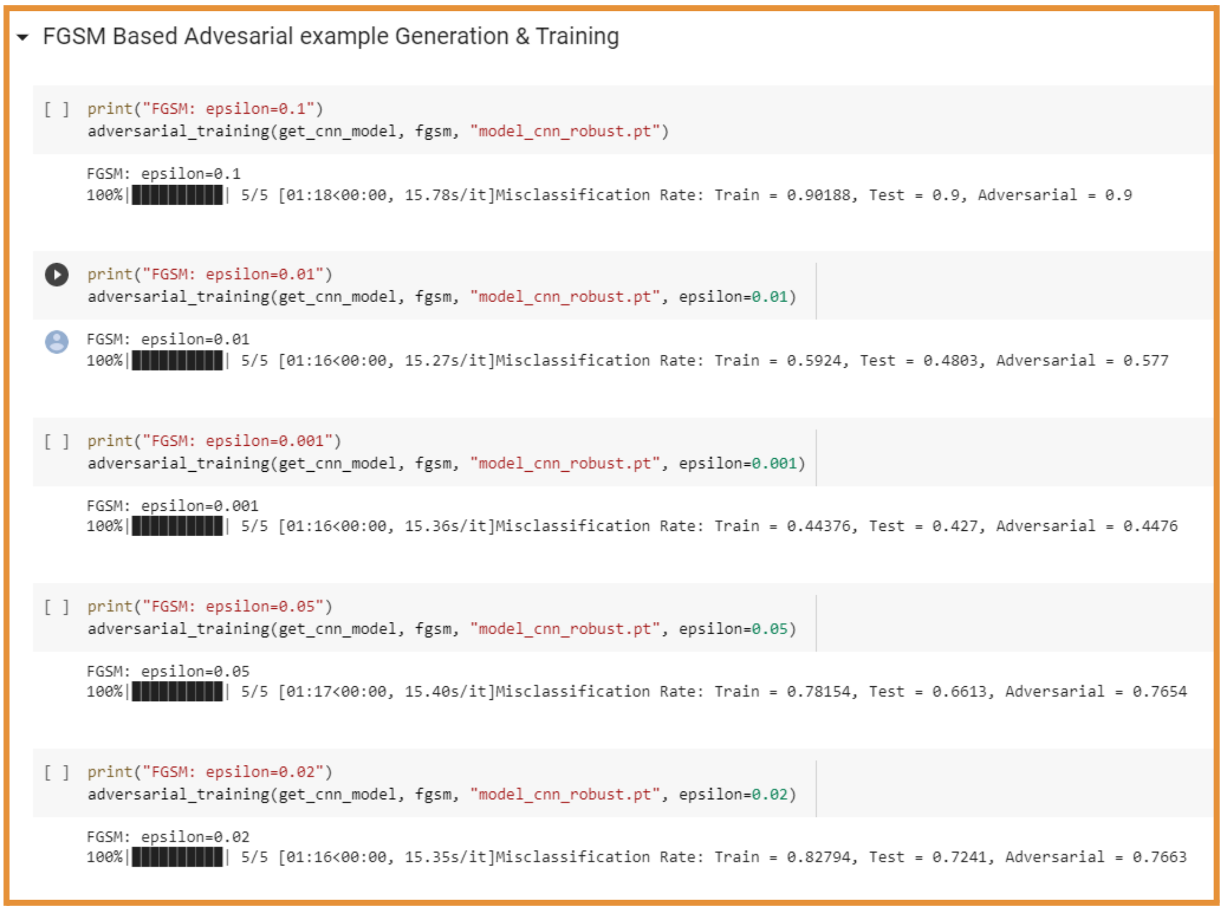
```
[ ] print("PGD_l2: alpha=0.1, epsilon=0.1, iters=20")
    adversarial_training(get_dnn_model, pgd_l2, "model_cnn_robust.pt", alpha=0.1, epsilon=0.1)

    PGD_l2: alpha=0.1, epsilon=0.1, iters=20
    100%|██████████| 5/5 [01:44<00:00, 20.99s/it]Misclassification Rate: Train = 0.04956666666666667, Test = 0.0382, Adversarial = 0.049
```

```
    print("PGD_l2: alpha=0.05, epsilon=0.05, iters=20")
    adversarial_training(get_dnn_model, pgd_l2, "model_cnn_robust.pt", alpha=0.05, epsilon=0.05)

    PGD_l2: alpha=0.05, epsilon=0.05, iters=20
    100%|██████████| 5/5 [01:44<00:00, 20.83s/it]Misclassification Rate: Train = 0.0465, Test = 0.0396, Adversarial = 0.0449
```

```
[ ] print("PGD_l2: alpha=0.001, epsilon=0.01, iters=20")
    adversarial_training(get_dnn_model, pgd_l2, "model_cnn_robust.pt", alpha=0.001, epsilon=0.01)

    PGD_l2: alpha=0.001, epsilon=0.01, iters=20
    100%|██████████| 5/5 [01:45<00:00, 21.11s/it]Misclassification Rate: Train = 0.04346666666666667, Test = 0.0413, Adversarial = 0.0419
```

```
[ ] print("PGD_l2: alpha=0.01, epsilon=0.1, iters=40")
    adversarial_training(get_dnn_model, pgd_l2, "model_cnn_robust.pt", alpha=0.01, epsilon=0.1, num_iter=40)

    PGD_l2: alpha=0.01, epsilon=0.1, iters=40
    100%|██████████| 5/5 [02:48<00:00, 33.75s/it]Misclassification Rate: Train = 0.04948333333333333, Test = 0.0383, Adversarial = 0.0494
```

3) CNN_CIFAR:



FGSM Based Advesarial example Generation & Training

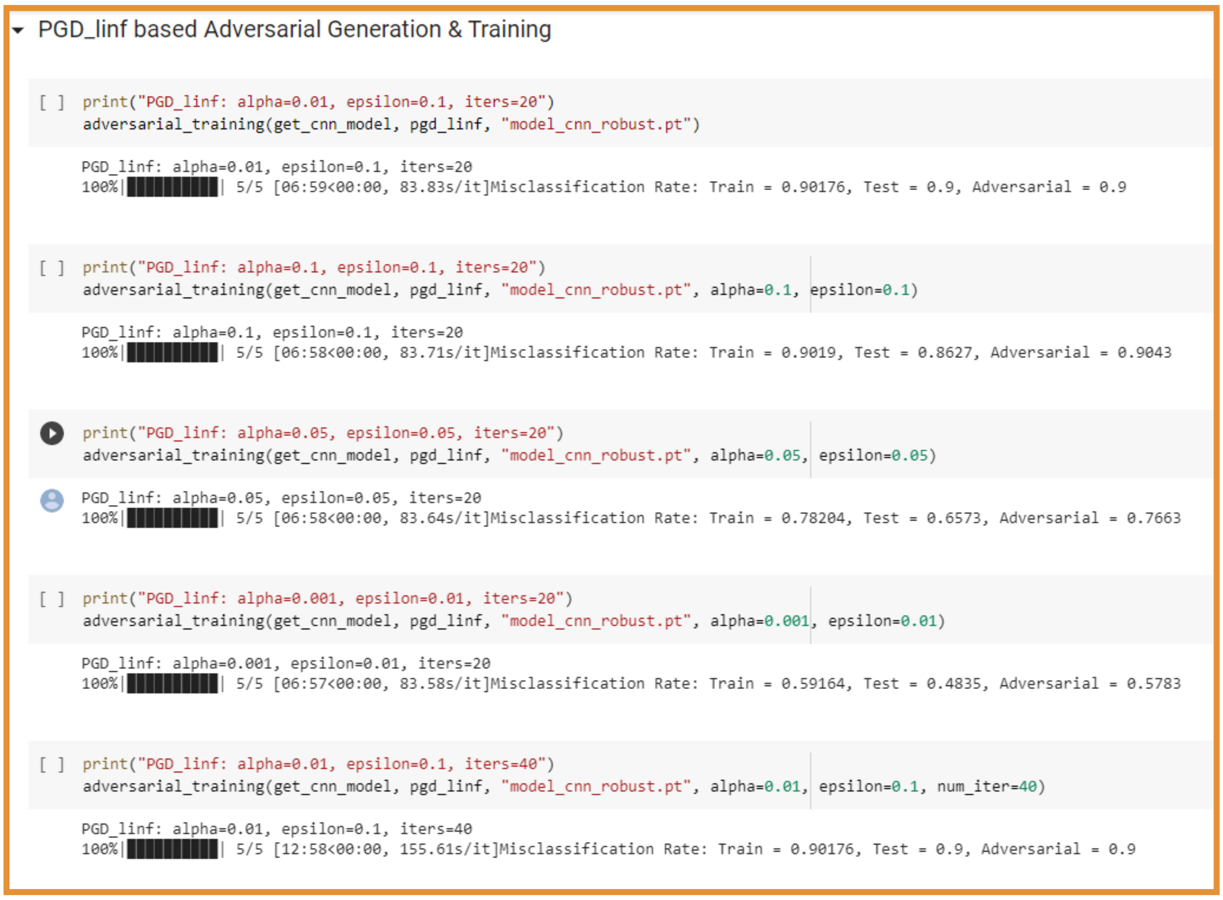
```
[ ] print("FGSM: epsilon=0.1")
    adversarial_training(get_cnn_model, fgsm, "model_cnn_robust.pt")

    FGSM: epsilon=0.1
    100%|██████████| 5/5 [01:18<00:00, 15.78s/it]Misclassification Rate: Train = 0.90188, Test = 0.9, Adversarial = 0.9
```

```
    print("FGSM: epsilon=0.01")
    adversarial_training(get_cnn_model, fgsm, "model_cnn_robust.pt", epsilon=0.01)

    FGSM: epsilon=0.01
    100%|██████████| 5/5 [01:16<00:00, 15.27s/it]Misclassification Rate: Train = 0.5924, Test = 0.4803, Adversarial = 0.577
```

```
[ ] print("FGSM: epsilon=0.001")
    adversarial_training(get_cnn_model, fgsm, "model_cnn_robust.pt", epsilon=0.001)

    FGSM: epsilon=0.001
    100%|██████████| 5/5 [01:16<00:00, 15.36s/it]Misclassification Rate: Train = 0.44376, Test = 0.427, Adversarial = 0.4476
```

```
[ ] print("FGSM: epsilon=0.05")
    adversarial_training(get_cnn_model, fgsm, "model_cnn_robust.pt", epsilon=0.05)

    FGSM: epsilon=0.05
    100%|██████████| 5/5 [01:17<00:00, 15.40s/it]Misclassification Rate: Train = 0.78154, Test = 0.6613, Adversarial = 0.7654
```

```
[ ] print("FGSM: epsilon=0.02")
    adversarial_training(get_cnn_model, fgsm, "model_cnn_robust.pt", epsilon=0.02)

    FGSM: epsilon=0.02
    100%|██████████| 5/5 [01:16<00:00, 15.35s/it]Misclassification Rate: Train = 0.82794, Test = 0.7241, Adversarial = 0.7663
```



PGD_linf based Adversarial Generation & Training

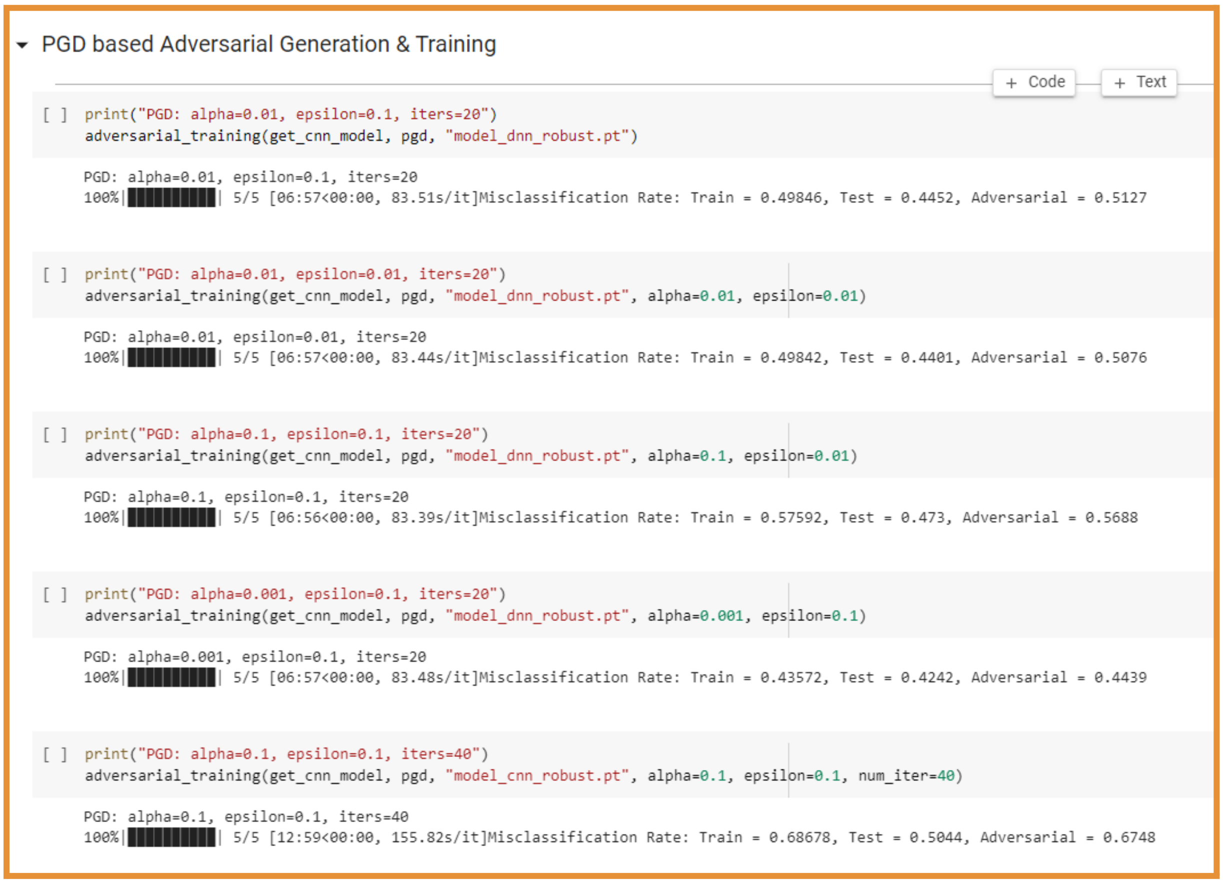
```
[ ] print("PGD_linf: alpha=0.01, epsilon=0.1, iters=20")
    adversarial_training(get_cnn_model, pgd_linf, "model_cnn_robust.pt")

    PGD_linf: alpha=0.01, epsilon=0.1, iters=20
    100%|██████████| 5/5 [06:59<00:00, 83.83s/it]Misclassification Rate: Train = 0.90176, Test = 0.9, Adversarial = 0.9
```

```
[ ] print("PGD_linf: alpha=0.1, epsilon=0.1, iters=20")
    adversarial_training(get_cnn_model, pgd_linf, "model_cnn_robust.pt", alpha=0.1, epsilon=0.1)

    PGD_linf: alpha=0.1, epsilon=0.1, iters=20
    100%|██████████| 5/5 [06:58<00:00, 83.71s/it]Misclassification Rate: Train = 0.9019, Test = 0.8627, Adversarial = 0.9043
```

```
    print("PGD_linf: alpha=0.05, epsilon=0.05, iters=20")
    adversarial_training(get_cnn_model, pgd_linf, "model_cnn_robust.pt", alpha=0.05, epsilon=0.05)

    PGD_linf: alpha=0.05, epsilon=0.05, iters=20
    100%|██████████| 5/5 [06:58<00:00, 83.64s/it]Misclassification Rate: Train = 0.78204, Test = 0.6573, Adversarial = 0.7663
```

```
[ ] print("PGD_linf: alpha=0.001, epsilon=0.01, iters=20")
    adversarial_training(get_cnn_model, pgd_linf, "model_cnn_robust.pt", alpha=0.001, epsilon=0.01)

    PGD_linf: alpha=0.001, epsilon=0.01, iters=20
    100%|██████████| 5/5 [06:57<00:00, 83.58s/it]Misclassification Rate: Train = 0.59164, Test = 0.4835, Adversarial = 0.5783
```

```
[ ] print("PGD_linf: alpha=0.01, epsilon=0.1, iters=40")
    adversarial_training(get_cnn_model, pgd_linf, "model_cnn_robust.pt", alpha=0.01, epsilon=0.1, num_iter=40)

    PGD_linf: alpha=0.01, epsilon=0.1, iters=40
    100%|██████████| 5/5 [12:58<00:00, 155.61s/it]Misclassification Rate: Train = 0.90176, Test = 0.9, Adversarial = 0.9
```



PGD based Adversarial Generation & Training

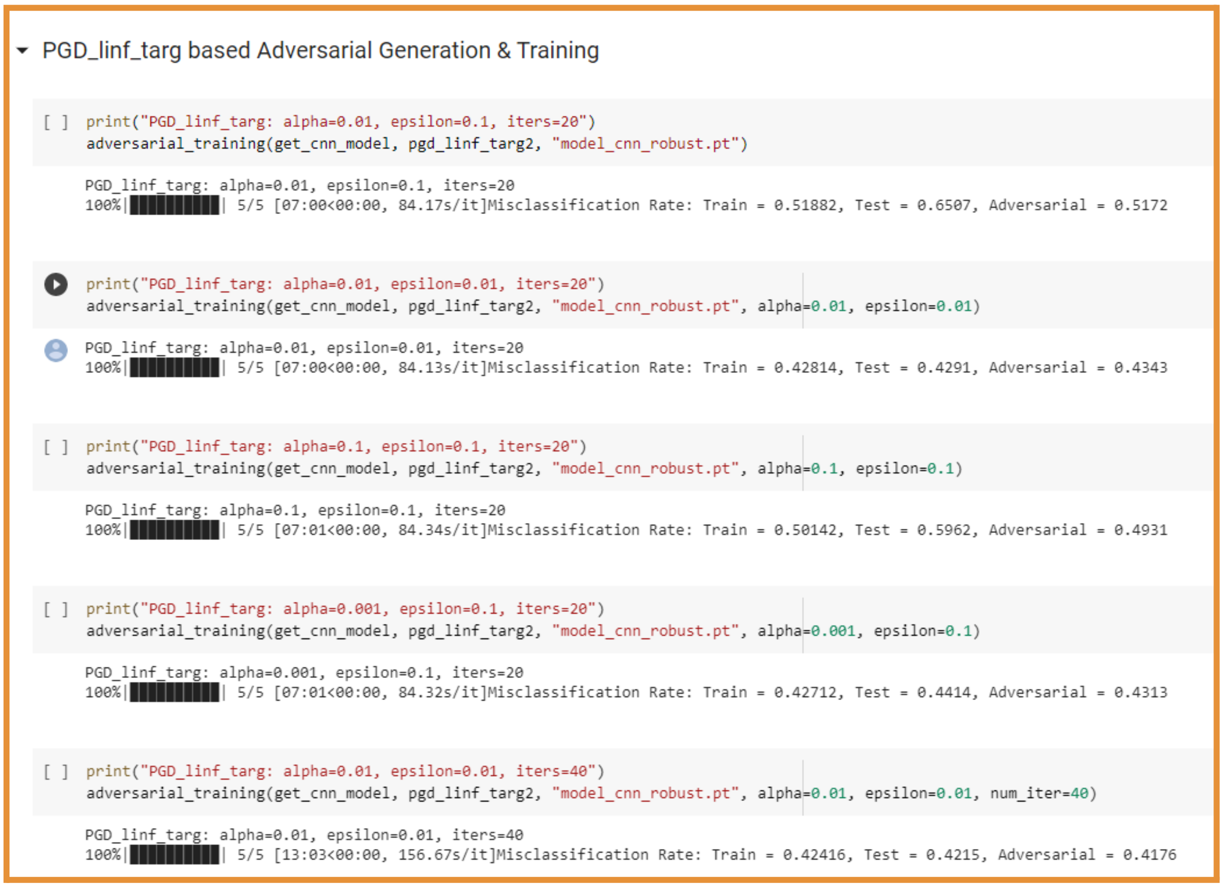
```
[ ] print("PGD: alpha=0.01, epsilon=0.1, iters=20")
    adversarial_training(get_cnn_model, pgd, "model_dnn_robust.pt")

    PGD: alpha=0.01, epsilon=0.1, iters=20
    100%|██████████| 5/5 [06:57<00:00, 83.51s/it]Misclassification Rate: Train = 0.49846, Test = 0.4452, Adversarial = 0.5127
```

```
[ ] print("PGD: alpha=0.01, epsilon=0.01, iters=20")
    adversarial_training(get_cnn_model, pgd, "model_dnn_robust.pt", alpha=0.01, epsilon=0.01)

    PGD: alpha=0.01, epsilon=0.01, iters=20
    100%|██████████| 5/5 [06:57<00:00, 83.44s/it]Misclassification Rate: Train = 0.49842, Test = 0.4401, Adversarial = 0.5076
```

```
[ ] print("PGD: alpha=0.1, epsilon=0.1, iters=20")
    adversarial_training(get_cnn_model, pgd, "model_dnn_robust.pt", alpha=0.1, epsilon=0.01)

    PGD: alpha=0.1, epsilon=0.1, iters=20
    100%|██████████| 5/5 [06:56<00:00, 83.39s/it]Misclassification Rate: Train = 0.57592, Test = 0.473, Adversarial = 0.5688
```

```
[ ] print("PGD: alpha=0.001, epsilon=0.1, iters=20")
    adversarial_training(get_cnn_model, pgd, "model_dnn_robust.pt", alpha=0.001, epsilon=0.1)

    PGD: alpha=0.001, epsilon=0.1, iters=20
    100%|██████████| 5/5 [06:57<00:00, 83.48s/it]Misclassification Rate: Train = 0.43572, Test = 0.4242, Adversarial = 0.4439
```

```
[ ] print("PGD: alpha=0.1, epsilon=0.1, iters=40")
    adversarial_training(get_cnn_model, pgd, "model_cnn_robust.pt", alpha=0.1, epsilon=0.1, num_iter=40)

    PGD: alpha=0.1, epsilon=0.1, iters=40
    100%|██████████| 5/5 [12:59<00:00, 155.82s/it]Misclassification Rate: Train = 0.68678, Test = 0.5044, Adversarial = 0.6748
```



PGD_linf_targ based Adversarial Generation & Training

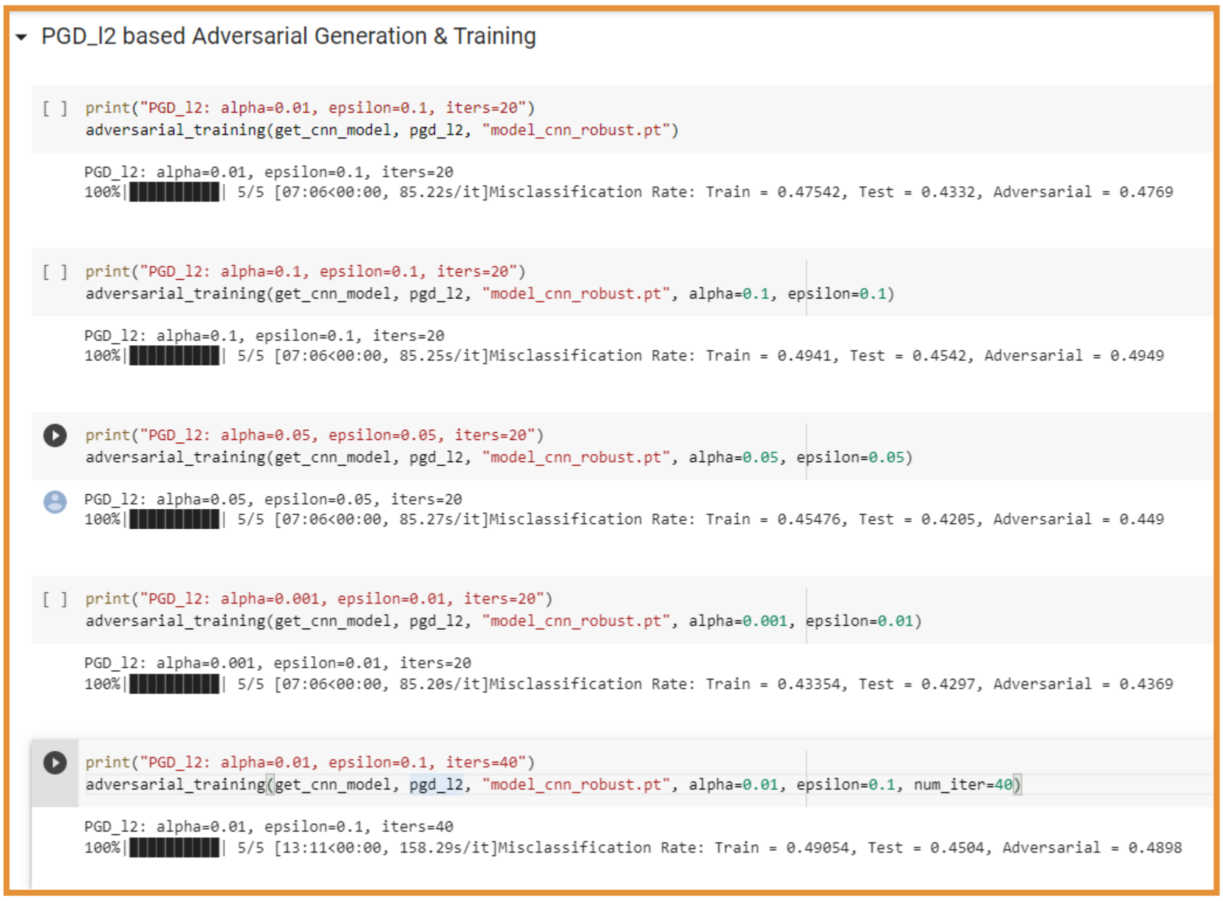
```
[ ] print("PGD_linf_targ: alpha=0.01, epsilon=0.1, iters=20")
    adversarial_training(get_cnn_model, pgd_linf_targ2, "model_cnn_robust.pt")

    PGD_linf_targ: alpha=0.01, epsilon=0.1, iters=20
    100%|██████████| 5/5 [07:00<00:00, 84.17s/it]Misclassification Rate: Train = 0.51882, Test = 0.6507, Adversarial = 0.5172
```

```
print("PGD_linf_targ: alpha=0.01, epsilon=0.01, iters=20")
adversarial_training(get_cnn_model, pgd_linf_targ2, "model_cnn_robust.pt", alpha=0.01, epsilon=0.01)

PGD_linf_targ: alpha=0.01, epsilon=0.01, iters=20
100%|██████████| 5/5 [07:00<00:00, 84.13s/it]Misclassification Rate: Train = 0.42814, Test = 0.4291, Adversarial = 0.4343
```

```
[ ] print("PGD_linf_targ: alpha=0.1, epsilon=0.1, iters=20")
    adversarial_training(get_cnn_model, pgd_linf_targ2, "model_cnn_robust.pt", alpha=0.1, epsilon=0.1)

    PGD_linf_targ: alpha=0.1, epsilon=0.1, iters=20
    100%|██████████| 5/5 [07:01<00:00, 84.34s/it]Misclassification Rate: Train = 0.50142, Test = 0.5962, Adversarial = 0.4931
```

```
[ ] print("PGD_linf_targ: alpha=0.001, epsilon=0.1, iters=20")
    adversarial_training(get_cnn_model, pgd_linf_targ2, "model_cnn_robust.pt", alpha=0.001, epsilon=0.1)

    PGD_linf_targ: alpha=0.001, epsilon=0.1, iters=20
    100%|██████████| 5/5 [07:01<00:00, 84.32s/it]Misclassification Rate: Train = 0.42712, Test = 0.4414, Adversarial = 0.4313
```

```
[ ] print("PGD_linf_targ: alpha=0.01, epsilon=0.01, iters=40")
    adversarial_training(get_cnn_model, pgd_linf_targ2, "model_cnn_robust.pt", alpha=0.01, epsilon=0.01, num_iter=40)

    PGD_linf_targ: alpha=0.01, epsilon=0.01, iters=40
    100%|██████████| 5/5 [13:03<00:00, 156.67s/it]Misclassification Rate: Train = 0.42416, Test = 0.4215, Adversarial = 0.4176
```



PGD_l2 based Adversarial Generation & Training

```
[ ] print("PGD_l2: alpha=0.01, epsilon=0.1, iters=20")
    adversarial_training(get_cnn_model, pgd_l2, "model_cnn_robust.pt")

    PGD_l2: alpha=0.01, epsilon=0.1, iters=20
    100%|██████████| 5/5 [07:06<00:00, 85.22s/it]Misclassification Rate: Train = 0.47542, Test = 0.4332, Adversarial = 0.4769
```

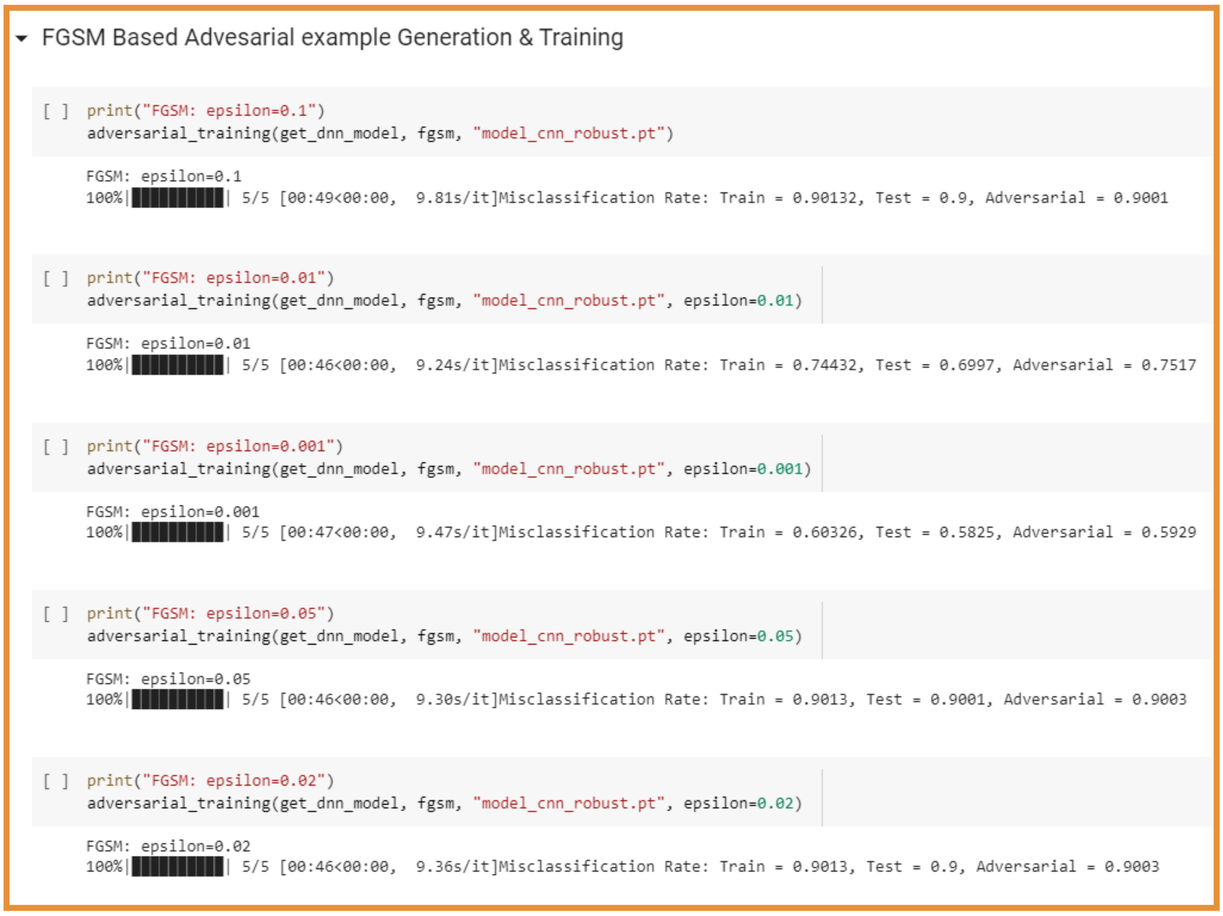
```
[ ] print("PGD_l2: alpha=0.1, epsilon=0.1, iters=20")
    adversarial_training(get_cnn_model, pgd_l2, "model_cnn_robust.pt", alpha=0.1, epsilon=0.1)

    PGD_l2: alpha=0.1, epsilon=0.1, iters=20
    100%|██████████| 5/5 [07:06<00:00, 85.25s/it]Misclassification Rate: Train = 0.4941, Test = 0.4542, Adversarial = 0.4949
```

```
print("PGD_l2: alpha=0.05, epsilon=0.05, iters=20")
adversarial_training(get_cnn_model, pgd_l2, "model_cnn_robust.pt", alpha=0.05, epsilon=0.05)

PGD_l2: alpha=0.05, epsilon=0.05, iters=20
100%|██████████| 5/5 [07:06<00:00, 85.27s/it]Misclassification Rate: Train = 0.45476, Test = 0.4205, Adversarial = 0.449
```

```
[ ] print("PGD_l2: alpha=0.001, epsilon=0.01, iters=20")
    adversarial_training(get_cnn_model, pgd_l2, "model_cnn_robust.pt", alpha=0.001, epsilon=0.01)

    PGD_l2: alpha=0.001, epsilon=0.01, iters=20
    100%|██████████| 5/5 [07:06<00:00, 85.20s/it]Misclassification Rate: Train = 0.43354, Test = 0.4297, Adversarial = 0.4369
```

```
print("PGD_l2: alpha=0.01, epsilon=0.1, iters=40")
adversarial_training(get_cnn_model, pgd_l2, "model_cnn_robust.pt", alpha=0.01, epsilon=0.1, num_iter=40)

PGD_l2: alpha=0.01, epsilon=0.1, iters=40
100%|██████████| 5/5 [13:11<00:00, 158.29s/it]Misclassification Rate: Train = 0.49054, Test = 0.4504, Adversarial = 0.4898
```

4) DNN_CIFAR:



FGSM Based Advesarial example Generation & Training

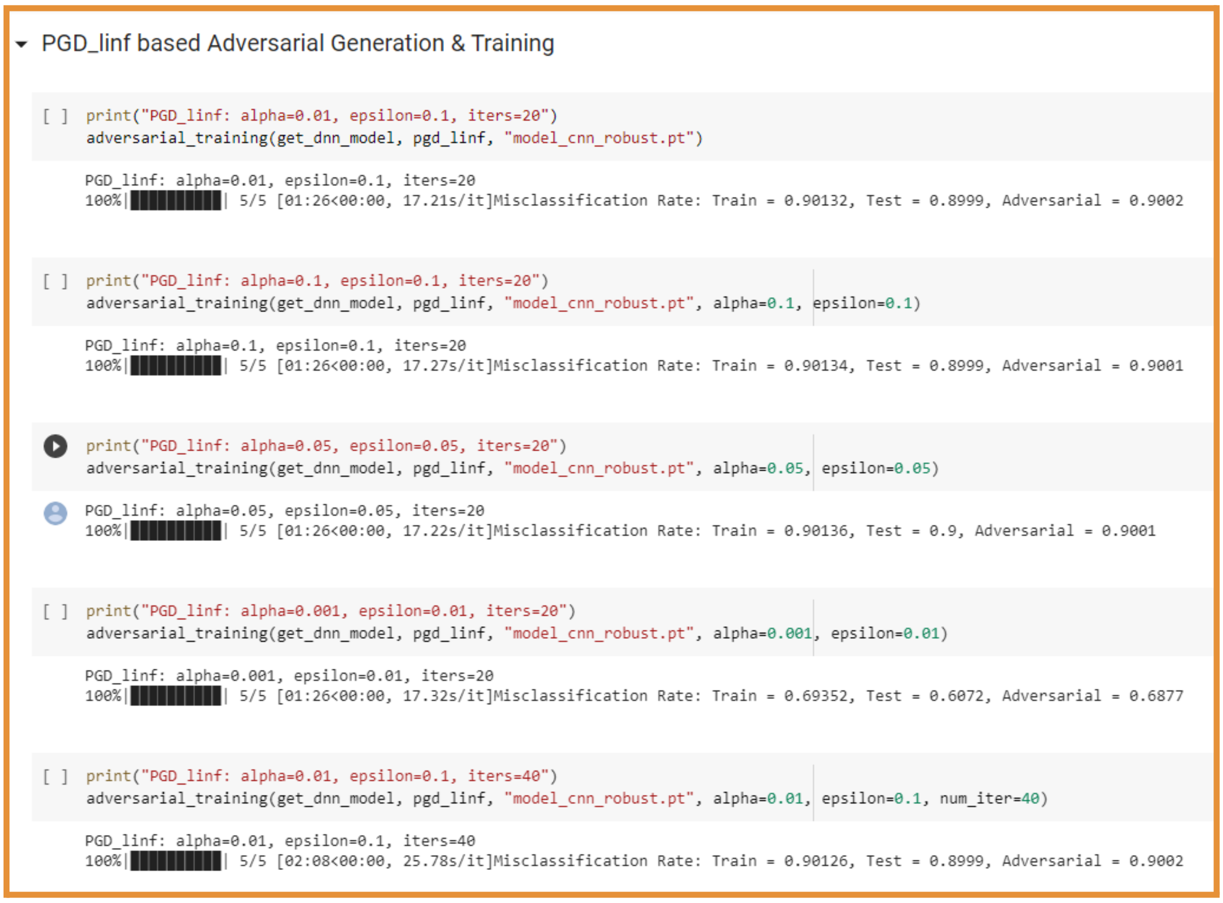
```
[ ] print("FGSM: epsilon=0.1")
    adversarial_training(get_dnn_model, fgsm, "model_cnn_robust.pt")

    FGSM: epsilon=0.1
    100%|██████████| 5/5 [00:49<00:00,  9.81s/it]Misclassification Rate: Train = 0.90132, Test = 0.9, Adversarial = 0.9001
```

```
[ ] print("FGSM: epsilon=0.01")
    adversarial_training(get_dnn_model, fgsm, "model_cnn_robust.pt", epsilon=0.01)

    FGSM: epsilon=0.01
    100%|██████████| 5/5 [00:46<00:00,  9.24s/it]Misclassification Rate: Train = 0.74432, Test = 0.6997, Adversarial = 0.7517
```

```
[ ] print("FGSM: epsilon=0.001")
    adversarial_training(get_dnn_model, fgsm, "model_cnn_robust.pt", epsilon=0.001)

    FGSM: epsilon=0.001
    100%|██████████| 5/5 [00:47<00:00,  9.47s/it]Misclassification Rate: Train = 0.60326, Test = 0.5825, Adversarial = 0.5929
```

```
[ ] print("FGSM: epsilon=0.05")
    adversarial_training(get_dnn_model, fgsm, "model_cnn_robust.pt", epsilon=0.05)

    FGSM: epsilon=0.05
    100%|██████████| 5/5 [00:46<00:00,  9.30s/it]Misclassification Rate: Train = 0.9013, Test = 0.9001, Adversarial = 0.9003
```

```
[ ] print("FGSM: epsilon=0.02")
    adversarial_training(get_dnn_model, fgsm, "model_cnn_robust.pt", epsilon=0.02)

    FGSM: epsilon=0.02
    100%|██████████| 5/5 [00:46<00:00,  9.36s/it]Misclassification Rate: Train = 0.9013, Test = 0.9, Adversarial = 0.9003
```



PGD_linf based Adversarial Generation & Training

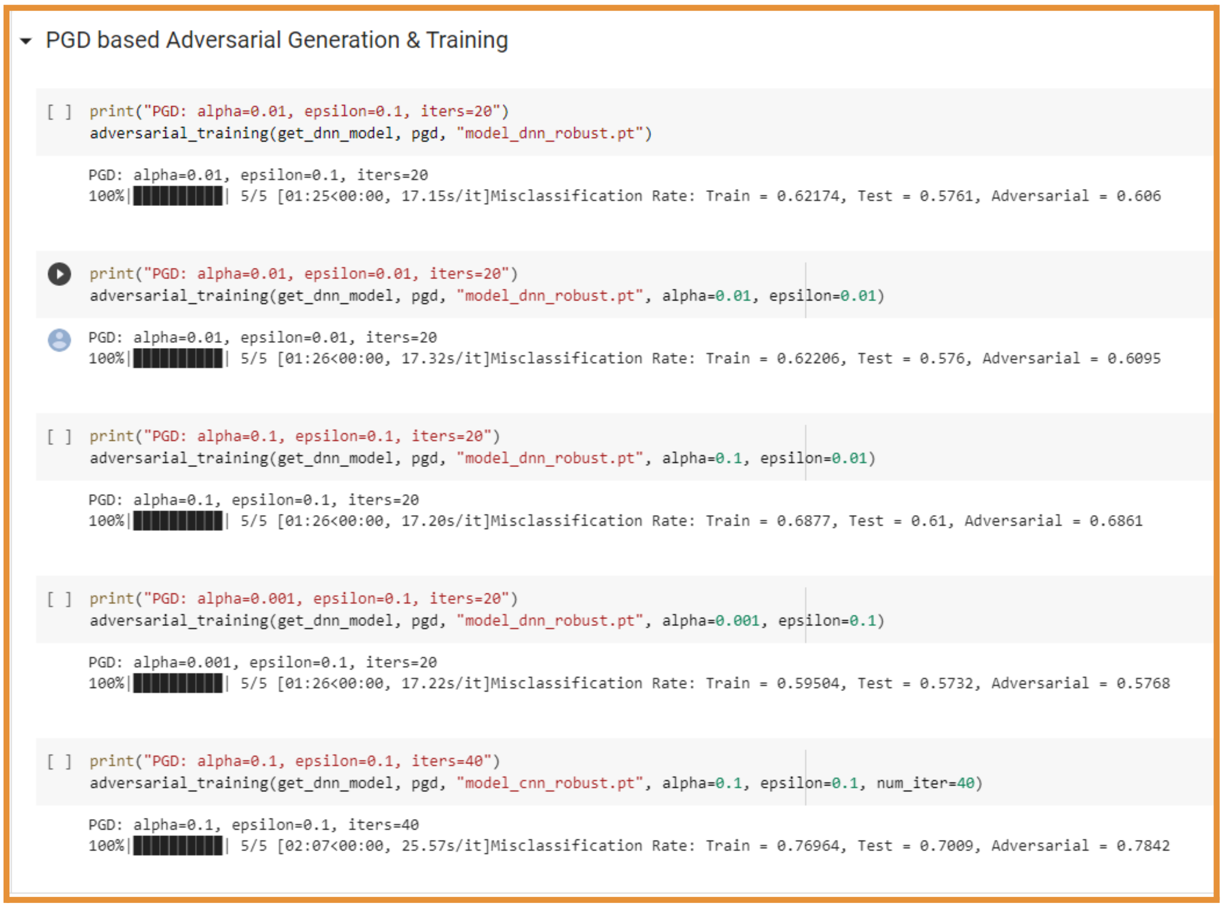
```
[ ] print("PGD_linf: alpha=0.01, epsilon=0.1, iters=20")
    adversarial_training(get_dnn_model, pgd_linf, "model_cnn_robust.pt")

    PGD_linf: alpha=0.01, epsilon=0.1, iters=20
    100%|██████████| 5/5 [01:26<00:00, 17.21s/it]Misclassification Rate: Train = 0.90132, Test = 0.8999, Adversarial = 0.9002
```

```
[ ] print("PGD_linf: alpha=0.1, epsilon=0.1, iters=20")
    adversarial_training(get_dnn_model, pgd_linf, "model_cnn_robust.pt", alpha=0.1, epsilon=0.1)

    PGD_linf: alpha=0.1, epsilon=0.1, iters=20
    100%|██████████| 5/5 [01:26<00:00, 17.27s/it]Misclassification Rate: Train = 0.90134, Test = 0.8999, Adversarial = 0.9001
```

```
    print("PGD_linf: alpha=0.05, epsilon=0.05, iters=20")
    adversarial_training(get_dnn_model, pgd_linf, "model_cnn_robust.pt", alpha=0.05, epsilon=0.05)

    PGD_linf: alpha=0.05, epsilon=0.05, iters=20
    100%|██████████| 5/5 [01:26<00:00, 17.22s/it]Misclassification Rate: Train = 0.90136, Test = 0.9, Adversarial = 0.9001
```

```
[ ] print("PGD_linf: alpha=0.001, epsilon=0.01, iters=20")
    adversarial_training(get_dnn_model, pgd_linf, "model_cnn_robust.pt", alpha=0.001, epsilon=0.01)

    PGD_linf: alpha=0.001, epsilon=0.01, iters=20
    100%|██████████| 5/5 [01:26<00:00, 17.32s/it]Misclassification Rate: Train = 0.69352, Test = 0.6072, Adversarial = 0.6877
```

```
[ ] print("PGD_linf: alpha=0.01, epsilon=0.1, iters=40")
    adversarial_training(get_dnn_model, pgd_linf, "model_cnn_robust.pt", alpha=0.01, epsilon=0.1, num_iter=40)

    PGD_linf: alpha=0.01, epsilon=0.1, iters=40
    100%|██████████| 5/5 [02:08<00:00, 25.78s/it]Misclassification Rate: Train = 0.90126, Test = 0.8999, Adversarial = 0.9002
```



PGD based Adversarial Generation & Training

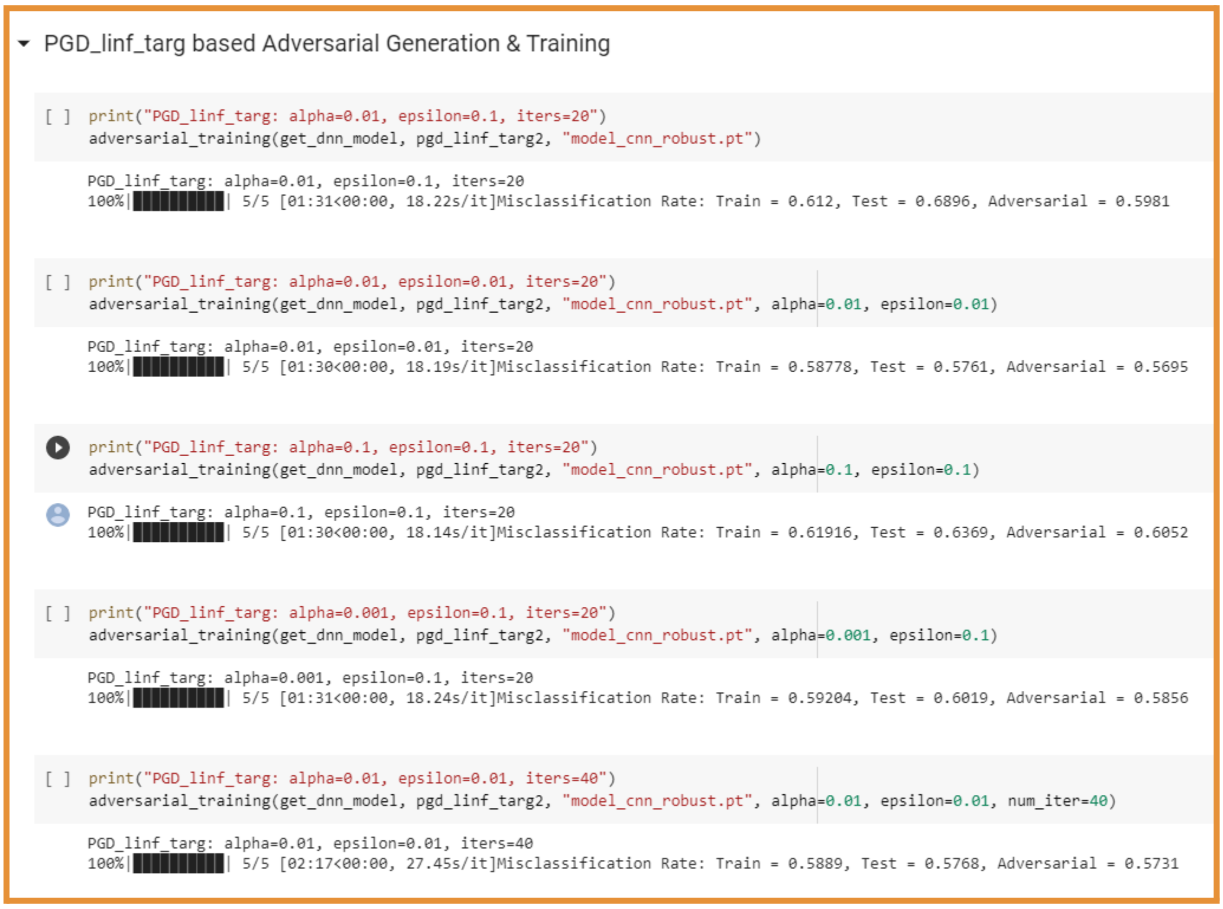
```
[ ] print("PGD: alpha=0.01, epsilon=0.1, iters=20")
    adversarial_training(get_dnn_model, pgd, "model_dnn_robust.pt")

    PGD: alpha=0.01, epsilon=0.1, iters=20
    100%|██████████| 5/5 [01:25<00:00, 17.15s/it]Misclassification Rate: Train = 0.62174, Test = 0.5761, Adversarial = 0.606
```

```
print("PGD: alpha=0.01, epsilon=0.01, iters=20")
adversarial_training(get_dnn_model, pgd, "model_dnn_robust.pt", alpha=0.01, epsilon=0.01)

PGD: alpha=0.01, epsilon=0.01, iters=20
100%|██████████| 5/5 [01:26<00:00, 17.32s/it]Misclassification Rate: Train = 0.62206, Test = 0.576, Adversarial = 0.6095
```

```
[ ] print("PGD: alpha=0.1, epsilon=0.1, iters=20")
    adversarial_training(get_dnn_model, pgd, "model_dnn_robust.pt", alpha=0.1, epsilon=0.01)

    PGD: alpha=0.1, epsilon=0.1, iters=20
    100%|██████████| 5/5 [01:26<00:00, 17.20s/it]Misclassification Rate: Train = 0.6877, Test = 0.61, Adversarial = 0.6861
```

```
[ ] print("PGD: alpha=0.001, epsilon=0.1, iters=20")
    adversarial_training(get_dnn_model, pgd, "model_dnn_robust.pt", alpha=0.001, epsilon=0.1)

    PGD: alpha=0.001, epsilon=0.1, iters=20
    100%|██████████| 5/5 [01:26<00:00, 17.22s/it]Misclassification Rate: Train = 0.59504, Test = 0.5732, Adversarial = 0.5768
```

```
[ ] print("PGD: alpha=0.1, epsilon=0.1, iters=40")
    adversarial_training(get_dnn_model, pgd, "model_cnn_robust.pt", alpha=0.1, epsilon=0.1, num_iter=40)

    PGD: alpha=0.1, epsilon=0.1, iters=40
    100%|██████████| 5/5 [02:07<00:00, 25.57s/it]Misclassification Rate: Train = 0.76964, Test = 0.7009, Adversarial = 0.7842
```



PGD_linf_targ based Adversarial Generation & Training

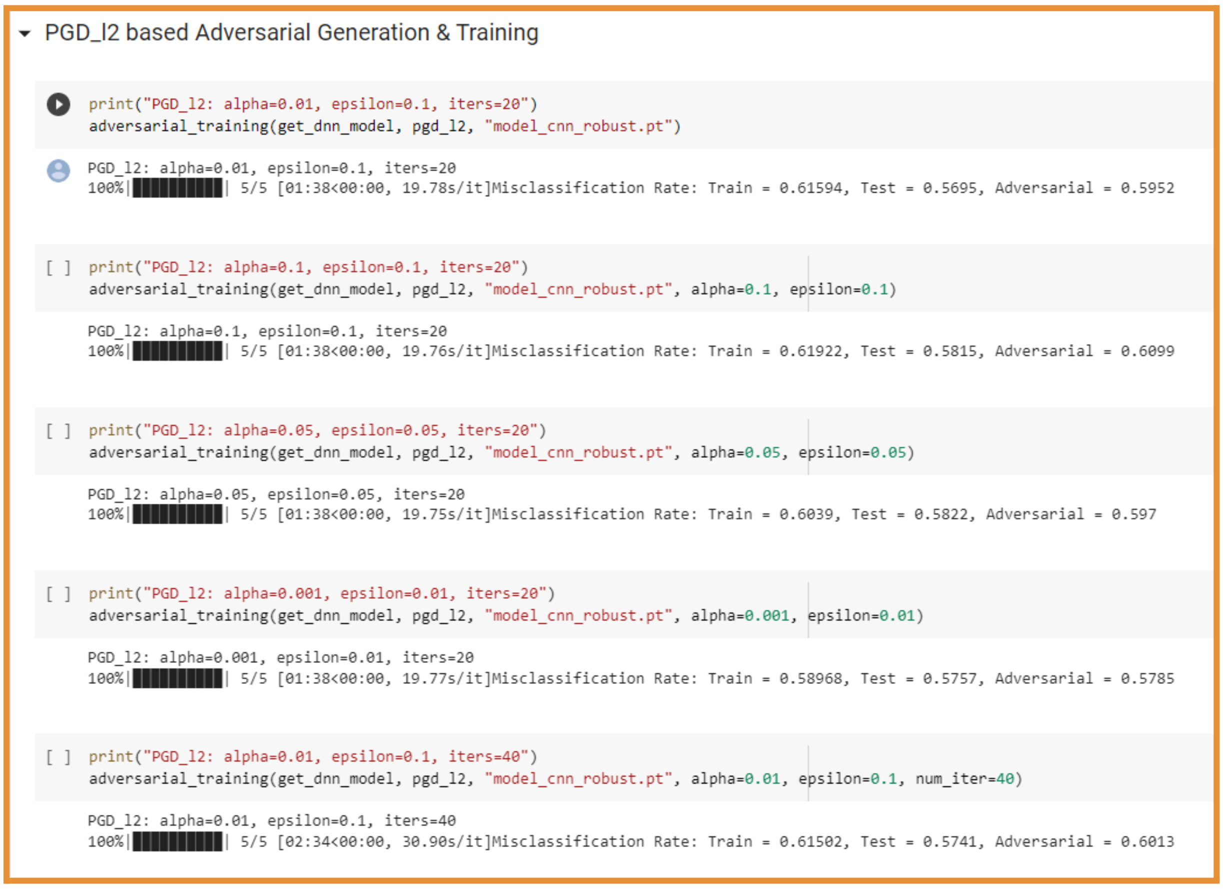
```
[ ] print("PGD_linf_targ: alpha=0.01, epsilon=0.1, iters=20")
    adversarial_training(get_dnn_model, pgd_linf_targ2, "model_cnn_robust.pt")

    PGD_linf_targ: alpha=0.01, epsilon=0.1, iters=20
    100%|██████████| 5/5 [01:31<00:00, 18.22s/it]Misclassification Rate: Train = 0.612, Test = 0.6896, Adversarial = 0.5981
```

```
[ ] print("PGD_linf_targ: alpha=0.01, epsilon=0.01, iters=20")
    adversarial_training(get_dnn_model, pgd_linf_targ2, "model_cnn_robust.pt", alpha=0.01, epsilon=0.01)

    PGD_linf_targ: alpha=0.01, epsilon=0.01, iters=20
    100%|██████████| 5/5 [01:30<00:00, 18.19s/it]Misclassification Rate: Train = 0.58778, Test = 0.5761, Adversarial = 0.5695
```

```
print("PGD_linf_targ: alpha=0.1, epsilon=0.1, iters=20")
adversarial_training(get_dnn_model, pgd_linf_targ2, "model_cnn_robust.pt", alpha=0.1, epsilon=0.1)

PGD_linf_targ: alpha=0.1, epsilon=0.1, iters=20
100%|██████████| 5/5 [01:30<00:00, 18.14s/it]Misclassification Rate: Train = 0.61916, Test = 0.6369, Adversarial = 0.6052
```

```
[ ] print("PGD_linf_targ: alpha=0.001, epsilon=0.1, iters=20")
    adversarial_training(get_dnn_model, pgd_linf_targ2, "model_cnn_robust.pt", alpha=0.001, epsilon=0.1)

    PGD_linf_targ: alpha=0.001, epsilon=0.1, iters=20
    100%|██████████| 5/5 [01:31<00:00, 18.24s/it]Misclassification Rate: Train = 0.59204, Test = 0.6019, Adversarial = 0.5856
```

```
[ ] print("PGD_linf_targ: alpha=0.01, epsilon=0.01, iters=40")
    adversarial_training(get_dnn_model, pgd_linf_targ2, "model_cnn_robust.pt", alpha=0.01, epsilon=0.01, num_iter=40)

    PGD_linf_targ: alpha=0.01, epsilon=0.01, iters=40
    100%|██████████| 5/5 [02:17<00:00, 27.45s/it]Misclassification Rate: Train = 0.5889, Test = 0.5768, Adversarial = 0.5731
```


PGD_l2 based Adversarial Generation & Training

```
print("PGD_l2: alpha=0.01, epsilon=0.1, iters=20")
adversarial_training(get_dnn_model, pgd_l2, "model_cnn_robust.pt")

PGD_l2: alpha=0.01, epsilon=0.1, iters=20
100%|██████████| 5/5 [01:38<00:00, 19.78s/it]Misclassification Rate: Train = 0.61594, Test = 0.5695, Adversarial = 0.5952
```

```
[ ] print("PGD_l2: alpha=0.1, epsilon=0.1, iters=20")
    adversarial_training(get_dnn_model, pgd_l2, "model_cnn_robust.pt", alpha=0.1, epsilon=0.1)

PGD_l2: alpha=0.1, epsilon=0.1, iters=20
100%|██████████| 5/5 [01:38<00:00, 19.76s/it]Misclassification Rate: Train = 0.61922, Test = 0.5815, Adversarial = 0.6099
```

```
[ ] print("PGD_l2: alpha=0.05, epsilon=0.05, iters=20")
    adversarial_training(get_dnn_model, pgd_l2, "model_cnn_robust.pt", alpha=0.05, epsilon=0.05)

PGD_l2: alpha=0.05, epsilon=0.05, iters=20
100%|██████████| 5/5 [01:38<00:00, 19.75s/it]Misclassification Rate: Train = 0.6039, Test = 0.5822, Adversarial = 0.597
```

```
[ ] print("PGD_l2: alpha=0.001, epsilon=0.01, iters=20")
    adversarial_training(get_dnn_model, pgd_l2, "model_cnn_robust.pt", alpha=0.001, epsilon=0.01)

PGD_l2: alpha=0.001, epsilon=0.01, iters=20
100%|██████████| 5/5 [01:38<00:00, 19.77s/it]Misclassification Rate: Train = 0.58968, Test = 0.5757, Adversarial = 0.5785
```

```
[ ] print("PGD_l2: alpha=0.01, epsilon=0.1, iters=40")
    adversarial_training(get_dnn_model, pgd_l2, "model_cnn_robust.pt", alpha=0.01, epsilon=0.1, num_iter=40)

PGD_l2: alpha=0.01, epsilon=0.1, iters=40
100%|██████████| 5/5 [02:34<00:00, 30.90s/it]Misclassification Rate: Train = 0.61502, Test = 0.5741, Adversarial = 0.6013
```

Conclusion:

The results for the adversarial training are depicted in the table for five different attacks. The best results for each setting are highlighted in green.