CSEC-720 Deep Learning Security (2225 Spring)

Experimental Assignment 2: Adversarial Examples

Zhi Liu[1] and Mehul Sen[1]

[1]Golisano College of Computing and Information Sciences, Rochester Institute of Technology

February 2023

1 Introduction

Adversarial examples are subtly modified inputs designed to deceive machine learning models. Adversarial training strengthens models by incorporating deceptive inputs during training. [1] We used a CNN model to classify handwritten digits using the MNIST dataset as a baseline and tested the FGSM and PGD adversarial attacks. We observed that FGSM attack effectiveness was greatest with epsilon values between 0.2 and 0.4. PGD attack, with proper alpha and iteration settings, demonstrated a stronger effect, reducing accuracy to 17% with a 0.2 epsilon. Adversarial training using FGSM and PGD enhanced the model's robustness [6], with retrained models performing better than new models trained on the combined dataset of benign and adversarial examples. [7] We also expected to see some decrease in accuracy when classifying the benign samples, but this did not occur. Surprisingly, PGD-trained models did not significantly improve in accuracy against FGSM attacks compared to baseline and FGSM-trained models. This may be due to the choice of PGD hyperparameters, label leaking [2], overfitting, or model capacity limitations [4].

2 Adversarial Examples

2.1 Baseline Model

The baseline model is a Convolutional Neural Network (CNN), is designed to classify handwritten digits utilizing the MNIST dataset. It consists of three convolutional layers, three max-pooling layers, and three fully connected layers. A detailed summary of the hyperparameters used in the model can be found in Table 1. With the MNIST testing dataset, this model achieved an 99% accuracy. For this assignment, the model was employed for both adversarial attack testing and adversarial training.

Hyperparameters	Baseline Model
Input Units	1x28x28
Layers	[3 Conv2D, 3 Pool, 3 FC]
Kernel Size	[2x2, 1x1]
# of Kernels	[32, 64, 128]
Pool Size	2x2
Batch Size	128
Loss Function	Cross Entropy Loss
Optimization Function	Adam
Activation Function	ReLU
Batch Normalization	[128, 64]
Learning Rate	0.001
Dropout Layers	0.5
Epochs	≥ 50

Table 1: Table of the Baseline Model Hyperparameters

In this assignment, we targeted the baseline model with FGSM and PGD attacks. Further details about these methods are discussed in the subsequent subsections.

2.2 FGSM Attack

The Fast Gradient Sign Method (FGSM) Attack [1], introduced by Goodfellow et al., is an adversarial attack technique primarily aimed at generating visually imperceptible adversarial examples that trick image classifiers into misclassifying images. The method involves the following four steps:

- Compute the gradient of the loss function with respect to the input image.
- Calculate the perturbation by taking the sign of the gradient and multiplying it by a small constant (epsilon).
- Generate the adversarial example by adding the perturbation to the original image.
- Feed the adversarial example into the targeted model to induce misclassification of the perturbed image.

We adapted the attack from PyTorch and perturbed the test images using epsilon values of 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, and 0.9. These perturbed images were then subject to misclassification as part of the FGSM attack. Although the baseline model was trained in batches of 128 image samples, we modified the test loader to handle individual images perturbed through the FGSM attack, with the goal of inducing misclassification.

Figure 1 shows the accuracy over epochs for the FGSM attack on the baseline model. We observe that as the epsilon value increases, the accuracy of the image classifier declines until it reaches a certain point, beyond which the accuracy remains unchanged. An epsilon value of 0.2 has the most significant impact on the accuracy of the image classifier, reducing its accuracy from 79% to 24%. Beyond an epsilon value of 0.4, the model no longer experiences additional accuracy loss and maintains a consistent accuracy of 10%. As a result, the most effective epsilon values for this attack on the baseline model lie between 0.2 and 0.4; increasing the epsilon value further does not impact the accuracy of the baseline model.

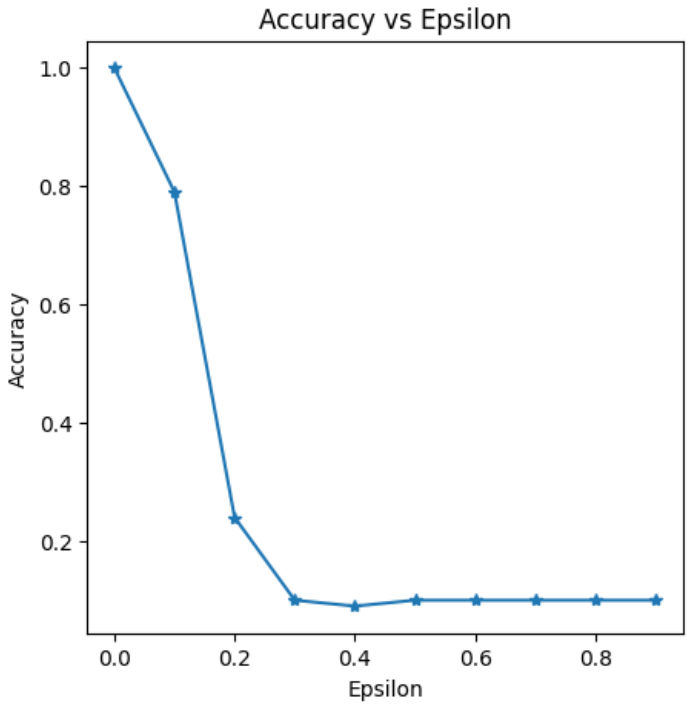

Figure 1: Accuracy over Epsilon for FGSM Adversarial Attack on the Baseline Model

Figure 2 displays 10 image samples from class '5' of the MNIST dataset, along with the perturbations introduced by the FGSM attack. As the epsilon value increases, the level of perturbation in the image intensifies. An intriguing observation is that initially, when the epsilon value is 0.2, the image is misclassified as belonging to class '2'. Subsequently, any increase in the epsilon value results in the image being misclassified as class '8'.

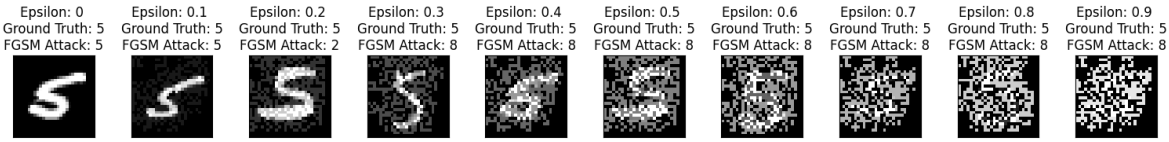

Figure 2: 10 Images showcasing the FGSM Adversarial Attack on the Baseline Model

2.3 PGD Attack

The Projected Gradient Descent (PGD) attack is a first-order attack against deep learning models, introduced in [4]. It works by computing the gradient of the loss function for the input data and then updating the input data in the direction of the gradient until the model misclassifies the data. Unlike the FGSM attack, the PGD iterates the gradient calculation process multiple times, taking small steps to generate adversarial samples with perturbations that maximize the model's loss function.

The advantage of the PGD attack compared to the FGSM attack is its universality. The PGD attack typically performs better against non-linear models, and models trained with adversarial samples generated by the PGD attack are generally more robust against other gradient-based attacks.

There are three major parameters for a PGD attack: epsilon, alpha, and the number of iterations. The epsilon value represents the range of perturbation in the input sample. The alpha value represents the step size of each iteration. The number of iterations represents how many attack iterations will be applied to one input sample.

In this assignment, the search range of epsilon is [0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]. The original research in [4] used an alpha value of 2/255 with 40 iterations. However, during my testing, I found that generating adversarial examples with this setting decreased the model accuracy to 0%, even with an epsilon value of 0.1. Although the attack was successful, it was very time-consuming and did not provide clear insights into how changing epsilon values would affect accuracy.

To better understand the trend of accuracy vs. epsilon, we tested multiple different alpha and iteration number settings. I found that using an alpha value of 0.5 and an iteration value of 10 produced results that better demonstrated the relationship between accuracy and epsilon.

Figure 3 shows the results of the PGD attack using adversarial samples generated under different epsilon settings against a model trained only with benign samples. The figure demonstrates that even with a low perturbation level of 0.2, the PGD attack can significantly decrease the model's performance to 17%. Furthermore, unlike the FGSM attack, increasing the epsilon value in the PGD attack can lead to further decreases in model accuracy. When epsilon exceeds 0.3, the classifier's accuracy is no more than 10%.

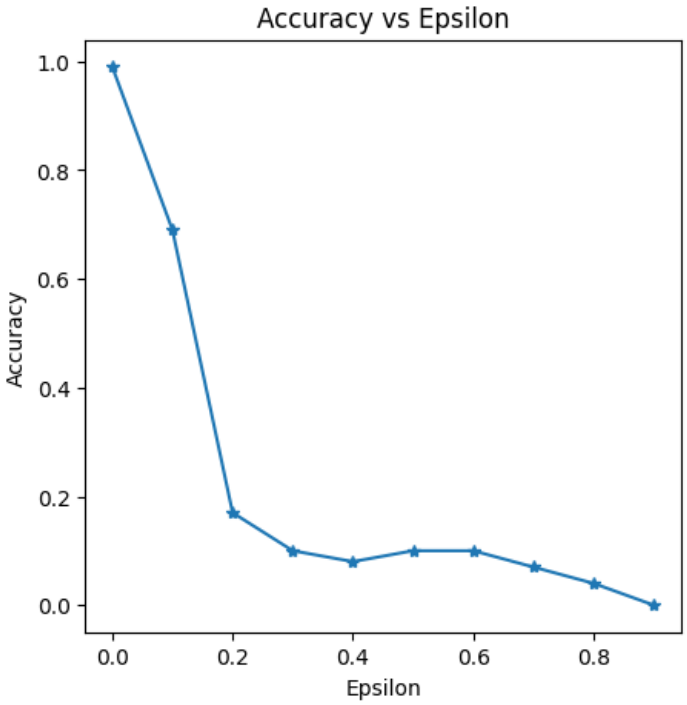

Figure 3: Accuracy over Epsilon for PGD Adversarial Attack on the Baseline Model

Figure 4 presents the perturbed images under different epsilon settings and demonstrates how they are misclassified by the classifier. The figure indicates that all of the misclassified samples are classified into the same class.

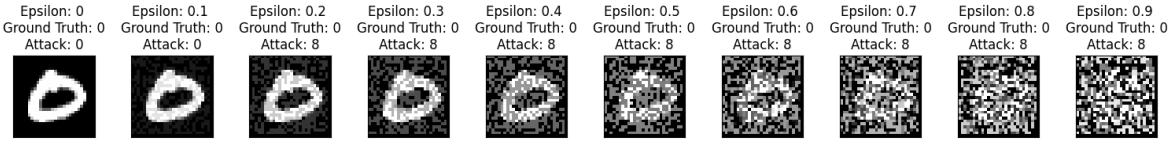

Figure 4: 10 Images showcasing the PGD Adversarial Attack on the Baseline Model

3 Adversarial Training

Adversarial training is a powerful defense against adversarial attacks. A widely adopted technique for adversarial training is iterative attack, which employs multiple attack iterations to generate adversarial examples [3, 4]. The goal of adversarial training is to enhance a neural network model's resilience against adversarial examples by training it with these examples. The stronger the adversarial examples used for training, the more robust the model becomes.

In this assignment, we train the baseline model using both FGSM and PGD attack techniques by iterating through several epsilon values.

3.1 Adversarial Training with FGSM

The adversarial training with FGSM proceeds through the following steps:

- **Generating Adversarial Examples.** Adversarial examples comprising perturbed images with epsilon values of 0.05, 0.1, 0.2, 0.25, and 0.3 are generated for the 60k training samples within the MNIST dataset.
- **Combine the Datasets.** The generated adversarial examples are then combined with the original training samples to create a new 'combined' training dataset, consisting of 360k images.

- **Training the Model.** The CNN model is subsequently trained on the combined training dataset.

For this assignment, we tested two FGSM adversarial training techniques: (1) Retraining the baseline model with the combined dataset (Retrained), and (2) Training a new model on the combined dataset (Untrained). Changing the architecture by adding additional layers or modifying the hyperparameters has been shown to have an effect on the accuracy of the adversarially trained model [4]. Therefore, to ensure that any changes in the accuracy of the model are solely dependent on the training of the model itself, we do not change the architecture or the hyperparameters of the model throughout this assignment.

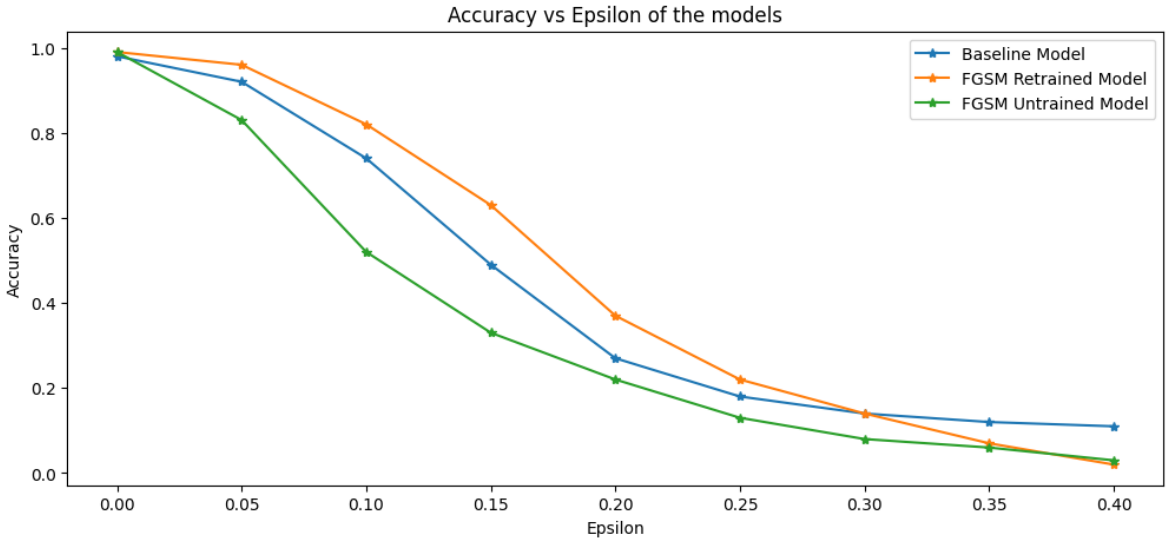

Figure 5: Accuracy vs Epsilon for Baseline, Original FGSM Trained (Retrained) and New FGSM Trained (Untrained) Models

Figure 5 displays the accuracy versus epsilon of three models: the baseline model, the original model trained using the combined dataset (retrained), and the new model trained on the combined dataset (untrained). We can see that the retrained model performs better than the baseline model, while the untrained model performs the worst. At approximately 0.30 epsilon, the accuracy of the baseline and the retrained model converges, while around 0.37 epsilon, the accuracy of the retrained model and the untrained model converges. Following this, we will use the retrained model as the default FGSM adversarially trained model. This model was able to achieve a 98% accuracy on the non-perturbed MNIST testing dataset.

3.2 Adversarial Training with PGD

The process for generating adversarial datasets using PGD is the same as that used for FGSM. For each epsilon value of [0.05, 0.1, 0.2, 0.25, 0.3], 60k adversarial samples were generated, along with 60k benign training samples used to train the baseline model, resulting in a total of 360k samples. These samples were used to train two models, one by retraining the baseline model that was trained using only benign samples, and the other by training a completely new model. The hyperparameters for both training processes were kept the same as the baseline model.

Both models were tested using the benign test set that was used to test the baseline model, as well as the previously generated 100 PGD adversarial samples.

When testing with the benign set, the retrained model achieved an accuracy of 98%, and the new model achieved an accuracy of 99%, which was no different from the performance of the benign model on the benign set.

Figure 6 displays the performance of the two adversarially trained models against PGD adversarial samples with different epsilon values. Both models demonstrated robustness against the attack, achieving an accuracy of 100%-98%. However, the accuracy of the retrained model dropped to 92%

when epsilon was set to 0.9. When the epsilon value gets higher, both models show a decreasing trend in accuracy.

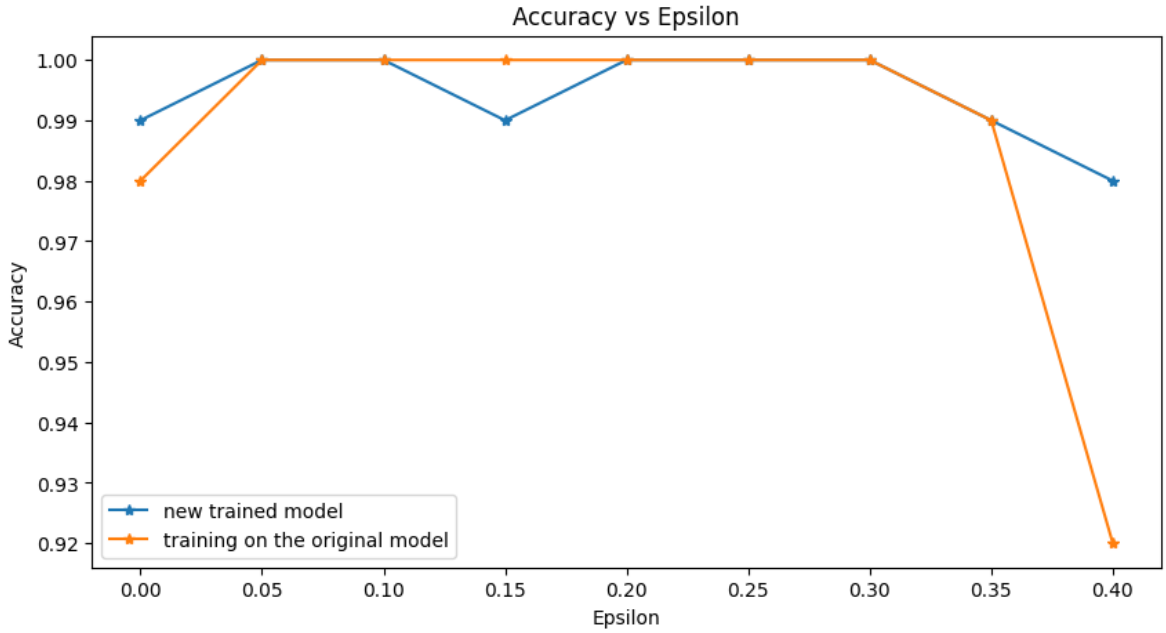

Figure 6: Accuracy vs Epsilon for both PGD adversarial trained models

3.3 Generalization of Adversarially Trained Models

We then tested how well the adversarially trained models generalized to each other. We first used the FGSM adversarially trained model against the PGD adversarial attack, and then we used the PGD adversarially trained model against the FGSM adversarial attack.

3.3.1 FGSM Adversarially Trained Model

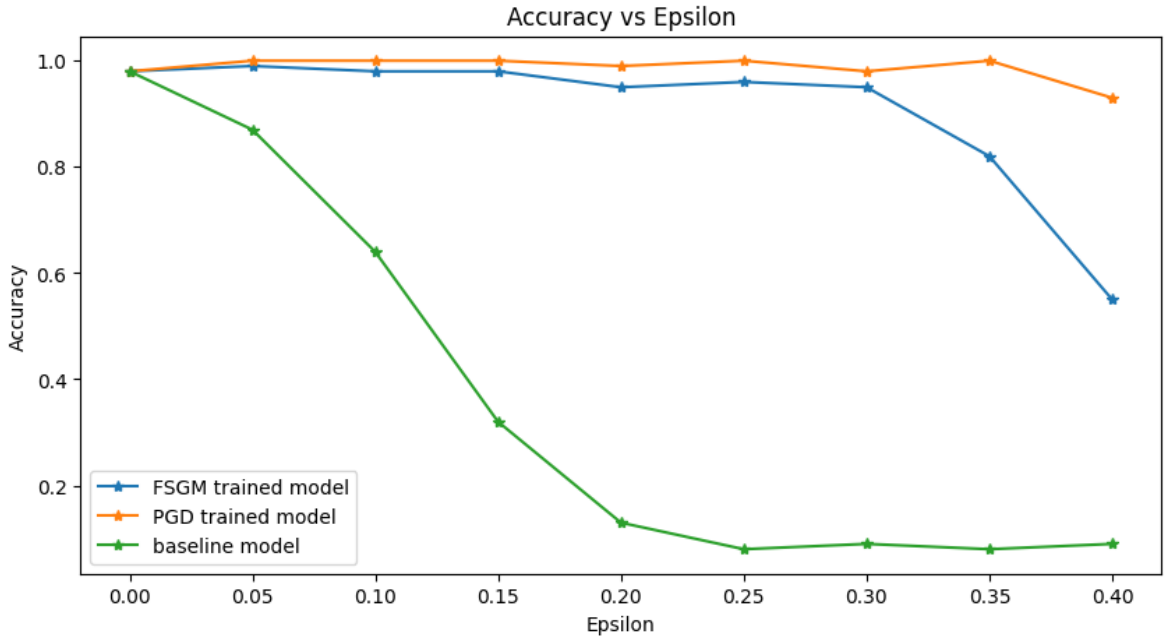

Figure 7: FGSM Adversarially Trained Model against PGD Adversarial Attack

Figure 7 displays the results of the PGD attack against the FGSM adversarially trained model compared to the baseline model. As can be seen, the FGSM-trained model shows a high level of robustness against the PGD attack compared to the baseline model. However, when the value of epsilon is increased, the accuracy performance of the FGSM-trained model eventually drops.

3.3.2 PGD Adversarially Trained Model

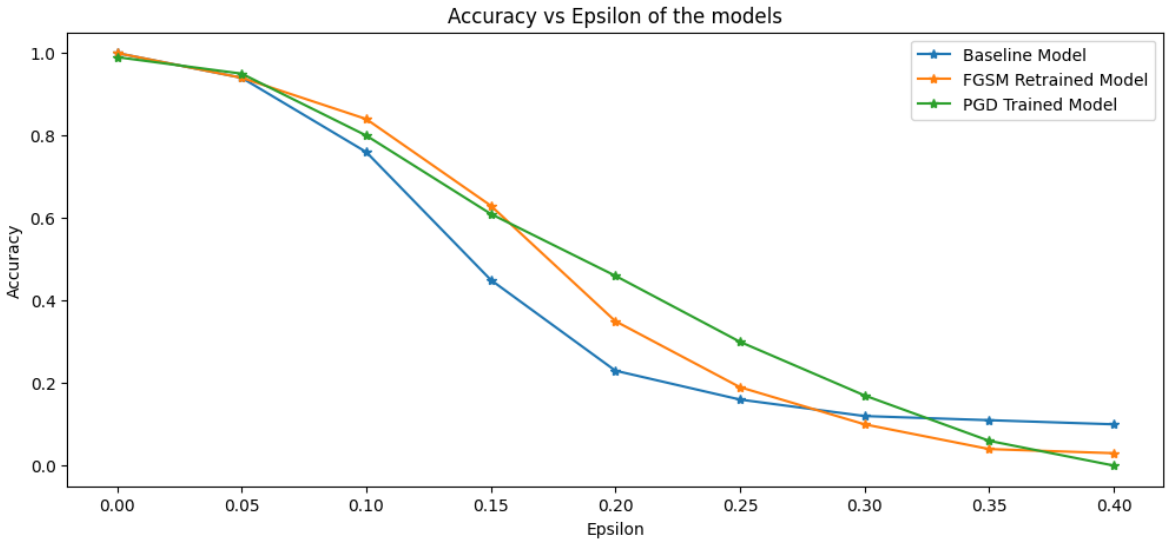

Figure 8: PGD Adversarially Trained Model against FGSM Adversarial Attack

Figure 8 shows the results of the FGSM adversarial attack on three models: the baseline model, the FGSM trained model, and the PGD trained model. We can see an increased robustness from the

baseline model in the PGD trained model up to 0.33 epsilon, after which the baseline model shows better results as compared to the adversarially trained models. The PGD model accuracy surpasses the FGSM trained model around 0.15 epsilon, after which it provides generally higher robustness as compared to the FGSM model.

Figure 9 shows ten perturbed image samples and their classifications by the baseline, FGSM Trained, and PGD Trained Models. We can see that the baseline model accurately predicts the class up to epsilon 0.15, the FGSM trained model accurately predicts the class up to epsilon 0.2, and the PGD trained model accurately predicts the class up to epsilon 0.25.

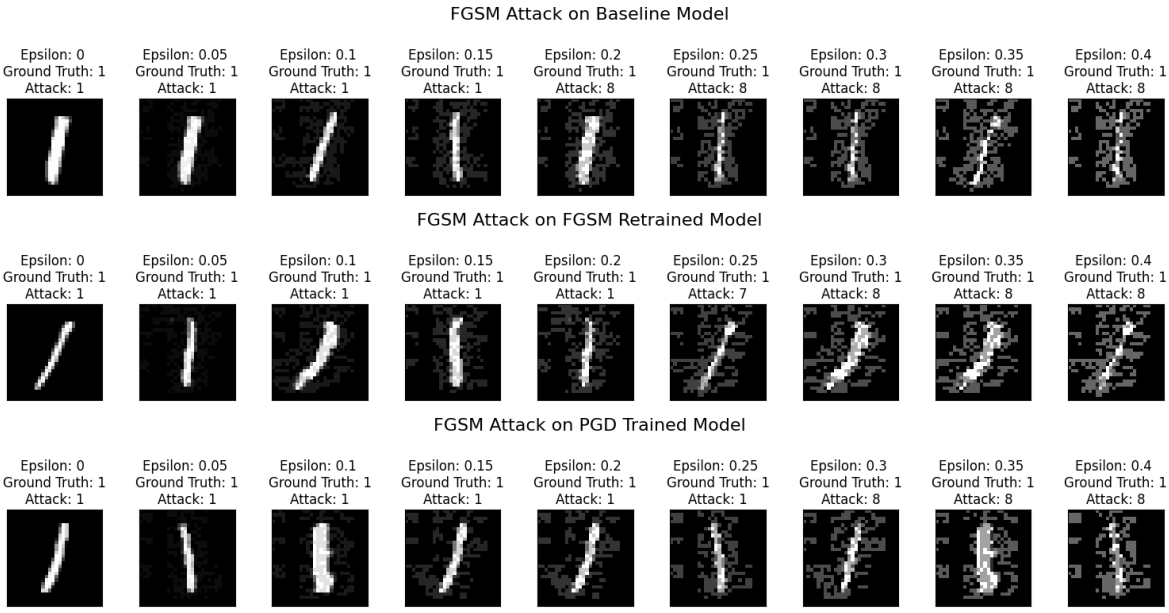

Figure 9: 10 Images showcasing the FGSM Adversarial Attack on the Baseline, FGSM Retrained and the PGD Trained Models

4 Analysis

Based on the results from the adversarial attack and training on the baseline CNN model, the following information is inferred:

4.1 Adversarial Examples

For both the FGSM and PGD adversarial attack scenarios, the maximum range of perturbation (epsilon) has a strong influence on the effectiveness of the attack. However, the relationship between the epsilon value and model accuracy is not linear. When epsilon increases, the accuracy of the model decreases quickly at first. After a certain epsilon value, the model accuracy decrease becomes negligible, at around 10For PGD, there are two other parameters, alpha and iteration number. Our tests indicate that a small alpha value with a larger number of iterations can produce the most effective adversarial samples. In [4], an alpha value of 0.0089 with 40/100 iterations was used. However, we used a very large alpha value and a very small number of iterations. An interesting observation to note was that the adversarial examples using FGSM on the baseline model caused the model to misclassify all the perturbed images to belong to the same class (class '8'). This supports the argument made by Goodfellow et al. in [1] stating that adversarial examples are caused due to excessive linearity within the model.

4.2 Adversarial Training

Another interesting observation was that the FGSM models were slightly less accurate (1%) compared to the baseline model when predicting images from the non-perturbed dataset. This confirms the

findings of several papers that discuss the trade-off between robustness and accuracy in adversarial training [8, 5].

While both the FGSM Untrained model and the FGSM Retrained model showed a similar curve between the accuracy and epsilon value, the FGSM Retrained model performed better compared to the untrained and the baseline model. This could be the case because rather than completely relearning the features from the benign and adversarial datasets, it had already learned features from the benign dataset. This allowed it to fine-tune the training model on the combined dataset such that it could generalize better and become more robust to adversarial attacks. This idea was presented by Tramer et al., where they leverage pre-trained models to generate adversarial examples such that it improved robustness against adversarial attacks [7].

4.3 Generalization of Adversarially Trained Models

Both models trained using adversarial training show increased robustness against PGD and FGSM adversarial attacks. This was due to the transferability property of adversarial examples where the examples generated for one model can stay adversarial to another model, hence adversarial training against one attack provides a level of robustness against the other attack [6].

Although we expected adversarial training to increase robustness, we did not anticipate such a high level of robustness in PGD trained models against the PGD attack. Additionally, we expected a decrease in model accuracy when testing with benign samples, but this did not occur. The FGSM retrained model exhibited similar robustness to the PGD-trained model in both the FGSM and PGD attacks, which was not what we had expected. This may be due to the similar diversity of the adversarial examples, hyperparameter choice, or the models learning robust features from the adversarial training process.

Interestingly, both adversarial-trained models demonstrated a sudden decrease in accuracy when the epsilon value of the PGD attack passed a certain threshold. This phenomenon has also been reported in the original study [4]. The retrained model's faster decrease may be due to limited model capacity since it had already been trained with benign samples.

One of the most crucial findings in [4] was that PGD adversarial-trained models demonstrated extensive robustness against transfer attacks, particularly when compared with other adversarial training techniques. However, during our test, we found that the increase in accuracy against FGSM compared to the baseline and FGSM trained model was not as high as we had expected. We suspect that this may be due to several reasons.

Firstly, it may be because we intentionally decreased the performance of the PGD model by increasing the alpha and reducing the number of iterations, which may have affected its universality. Secondly, it may be due to label leaking introduced in [2] and overfitting (this also happens with FGSM models). We observed that the adversarial-trained models had very good performance after only two epochs of training, even though we trained them for 50 epochs. Lastly, the capacity of the model may also influence the robustness of adversarial training, as discussed in [4].

5 Appendix

ChatGPT, developed by OpenAI, was utilized to brainstorm and enhance the quality of the paper. Prompt Used: "Improve the grammar and flow of this paragraph"

References

[1] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. "Explaining and Harnessing Adversarial Examples". In: *International Conference on Learning Representations* (2015).

[2] Shachar Kaufman et al. "Leakage in data mining: Formulation, detection, and avoidance". In: *ACM Transactions on Knowledge Discovery from Data* (2012).

[3] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. "Adversarial Machine Learning at Scale". In: *International Conference on Learning Representations* (2017).

[4] Aleksander Madry et al. "Towards Deep Learning Models Resistant to Adversarial Attacks". In: *International Conference on Learning Representations* (2018).

[5] Aditi Raghunathan et al. “Understanding and Mitigating the Tradeoff Between Robustness and Accuracy”. In: *International Conference on Machine Learning* (2020).

[6] Christian Szegedy et al. “Intriguing properties of neural networks”. In: *International Conference on Learning Representations* (2014).

[7] Florian Tramèr et al. “Ensemble Adversarial Training: Attacks and Defenses”. In: *International Conference on Learning Representations* (2018).

[8] Dimitris Tsipras et al. “Robustness May Be at Odds with Accuracy”. In: *International Conference on Learning Representations* (2019).