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 deceive machine learning models with subtle modifications. Adversarial training bolsters models by incorporating these deceptive inputs [1]. We employed a CNN model to classify the MNIST dataset's handwritten digits, testing FGSM [1] and PGD [5] attacks. FGSM attacks were most effective with epsilon values between 0.2 and 0.4. PGD attacks with suitable alpha and iterations, had a stronger impact, lowering accuracy to 17% at a 0.2 epsilon. Adversarial training with FGSM and PGD improved model robustness [7]; retrained models outperformed those trained on a combined dataset of benign and adversarial examples [8]. While expecting a benign sample classification accuracy decrease, this did not happen. Surprisingly, PGD-trained models did not significantly improve accuracy against FGSM attacks compared to baseline and FGSM-trained models. This could be caused due to certain PGD hyperparameters, label leaking [3], overfitting, or model capacity limitations [5].

2 Adversarial Examples

2.1 Baseline Model

The baseline model is a Convolutional Neural Network (CNN), which 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 the 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 observed that as the epsilon value increases, the accuracy of the image classifier declined until it reached a certain point, beyond which the accuracy remained 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 lies between 0.2 and 0.4; increasing the epsilon value further does not impact the accuracy of the baseline model.

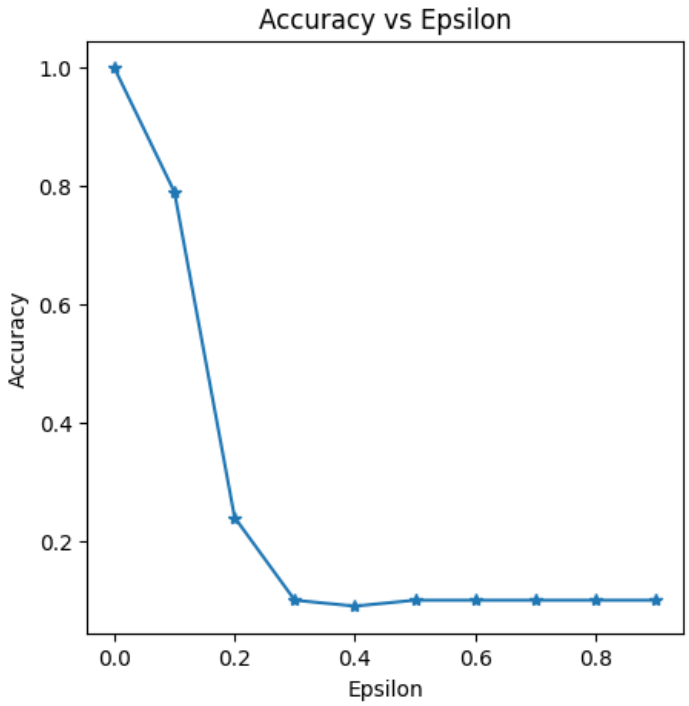

Figure 1: Accuracy over Epsilon for FGSM Adversarial Attack on the Baseline Model

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 increased, the level of perturbation in the image intensified. An intriguing observation was that initially, when the epsilon value was 0.2, the image was misclassified as belonging to class '2'. Subsequently, any increase in the epsilon value resulted in the image being misclassified as class '8'.

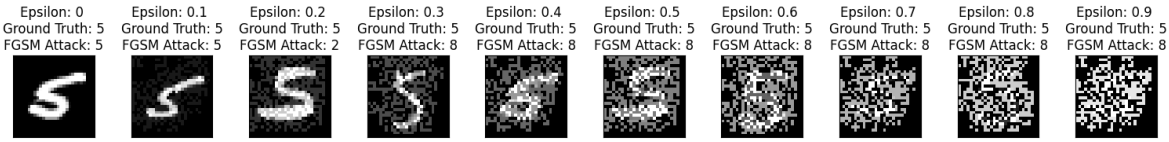

Figure 2: 10 Images showcasing the FGSM Adversarial Attack on the Baseline Model

2.3 PGD Attack

The Projected Gradient Descent (PGD) attack is a first-order attack against deep learning models, introduced in [5]. 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. According to [5], 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 [5] used an alpha value of 2/255 with 40 iterations. However, during our testing, we 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. We 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%.

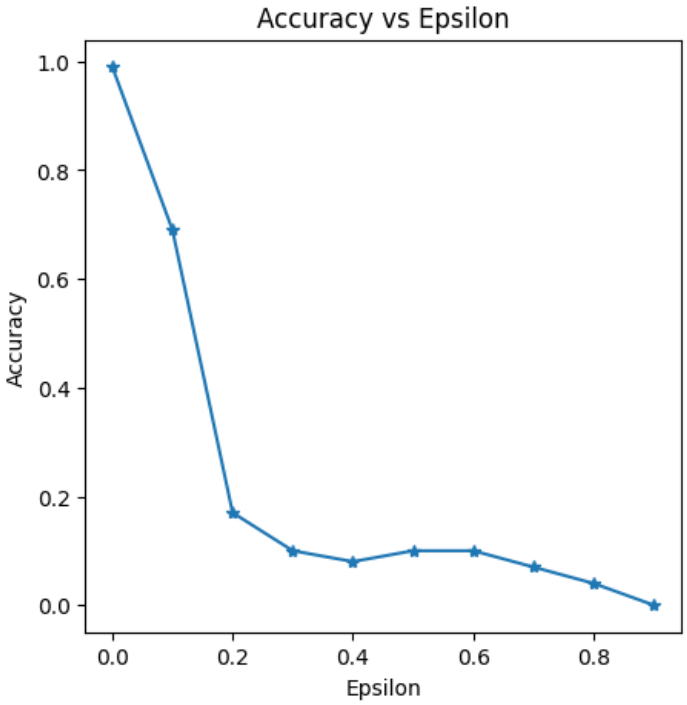

Figure 3: Accuracy over Epsilon for PGD Adversarial Attack on the Baseline Model

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.

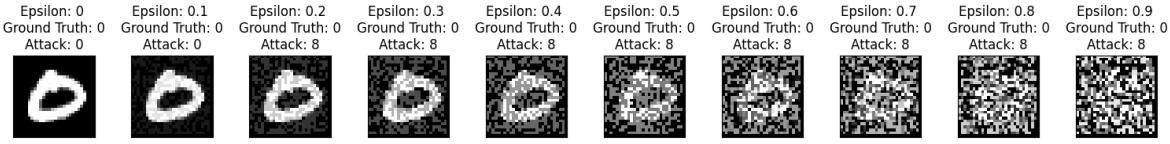

Figure 4: 10 Images showcasing the PGD Adversarial Attack on the Baseline Model

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 [4, 5]. 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.
- **Combining 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 [5]. 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.

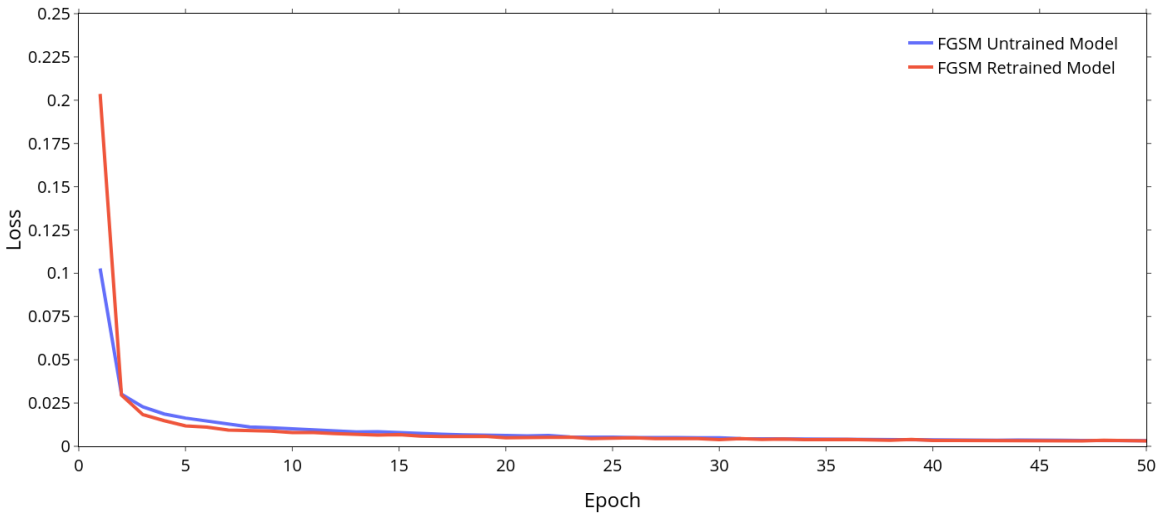

Figure 5: Loss vs Epoch for the Original FGSM Trained (Retrained) and the New FGSM Trained (Untrained) Models

Figure 5 displays the training loss over epochs for both the original model, retrained using the combined dataset, and the new model, trained on the combined dataset without prior knowledge. The untrained model's loss began at 0.102 for the first epoch and dropped to 0.002 by the fiftieth epoch. In contrast, the retrained model's loss started at 0.203 for the first epoch and decreased to 0.002 by the fiftieth epoch. This difference occurs because the new model learns about the dataset without prior information and can start with a lower loss than the original model. In contrast, the original model must relearn its classifications, resulting in almost double the initial loss. However, both models eventually reach the same loss over the fifty epochs, demonstrating their convergence.

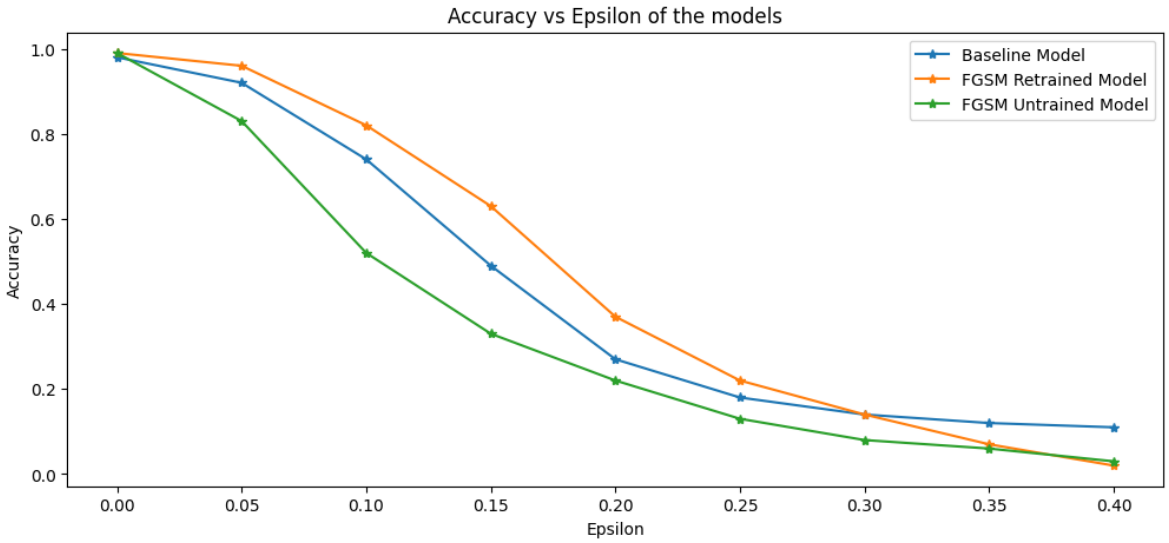

Figure 6: Accuracy vs Epsilon for Baseline, Original FGSM Trained (Retrained) and New FGSM Trained (Untrained) Models

Figure 6 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 performed better than the baseline model, while the untrained model performed the worst. At approximately 0.30 epsilon, the accuracy of the baseline and the retrained model converged, while around 0.37 epsilon, the accuracy of the retrained model and the untrained model converged. 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.

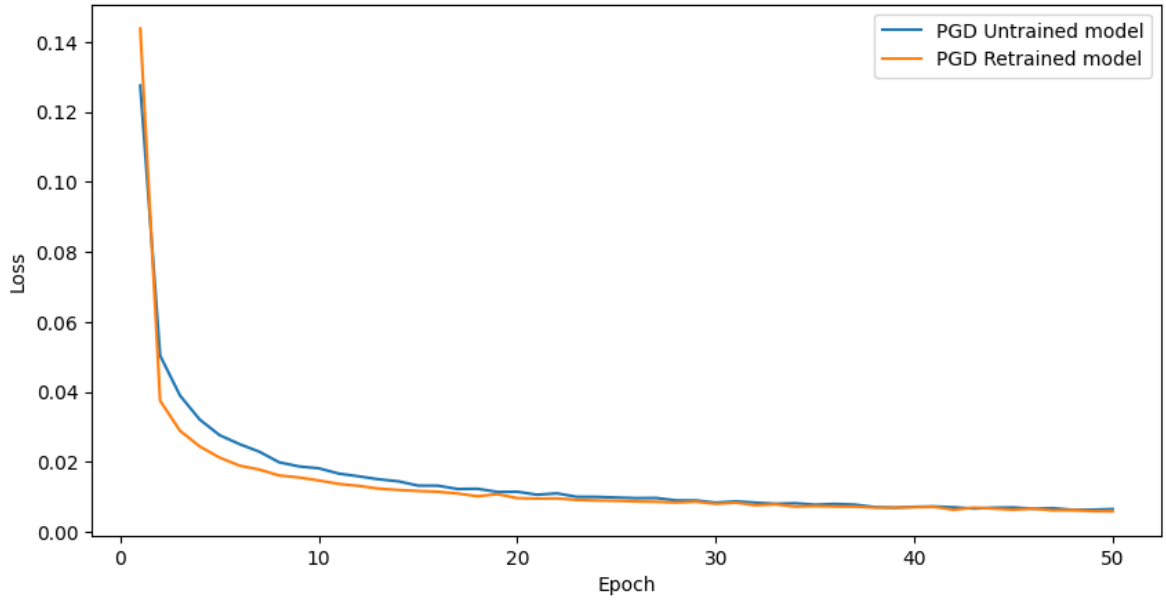

Figure 7: Loss vs Epoch for the Original PGD Trained (Retrained) and the New PGD Trained (Untrained) Models

Figure 7 shows the training loss vs epoch of both the untrained model and the retrained model with the sample generated above. The pattern of the training loss is the same as the FGSM-trained models. The retrained model has a higher training loss at first, but after that, the training loss is lower than the untrained model. We suspect that the reason for this is due to the adversarial sample aiming to maximize the loss of the already-trained model and make the loss generally higher than a completely new model. Additionally, the final training loss of the PGD-trained model is generally higher than the FGSM-trained 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 similar to the performance of the model trained on the benign set. The 1% difference in benign accuracy between the untrained and retrained models may be due to random weight initialization.

Figure 8 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. As the epsilon value increased, both models showed a reduction in their accuracy. The accuracy dropping rate of the trained model appears slower than the retrained one.

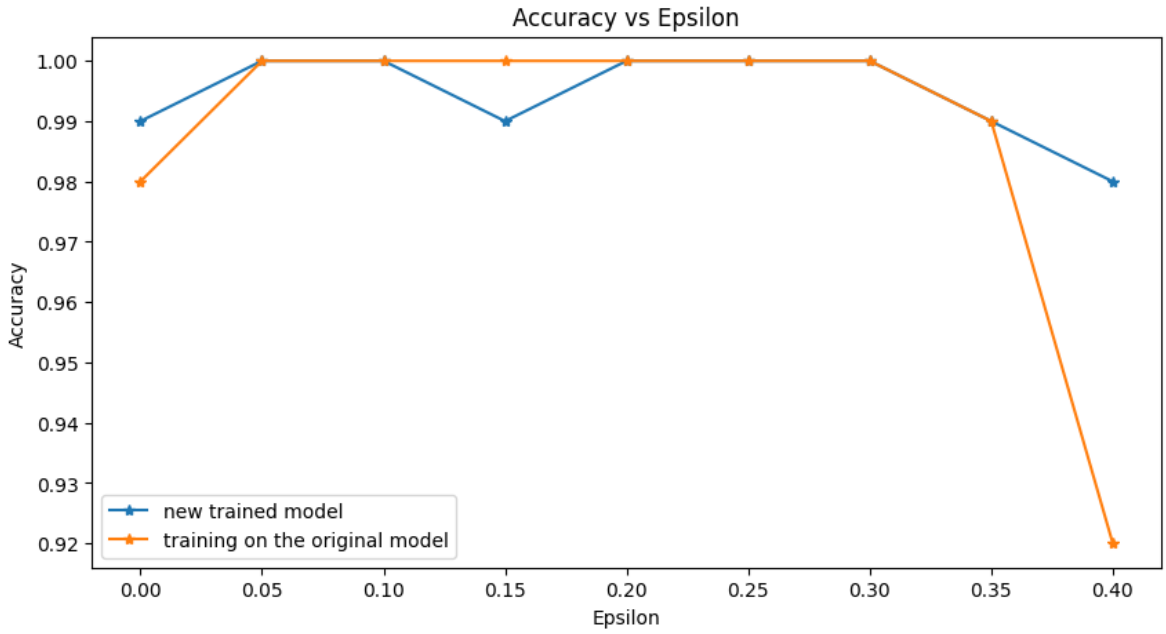

Figure 8: Accuracy vs Epsilon for both PGD adversarial trained models

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

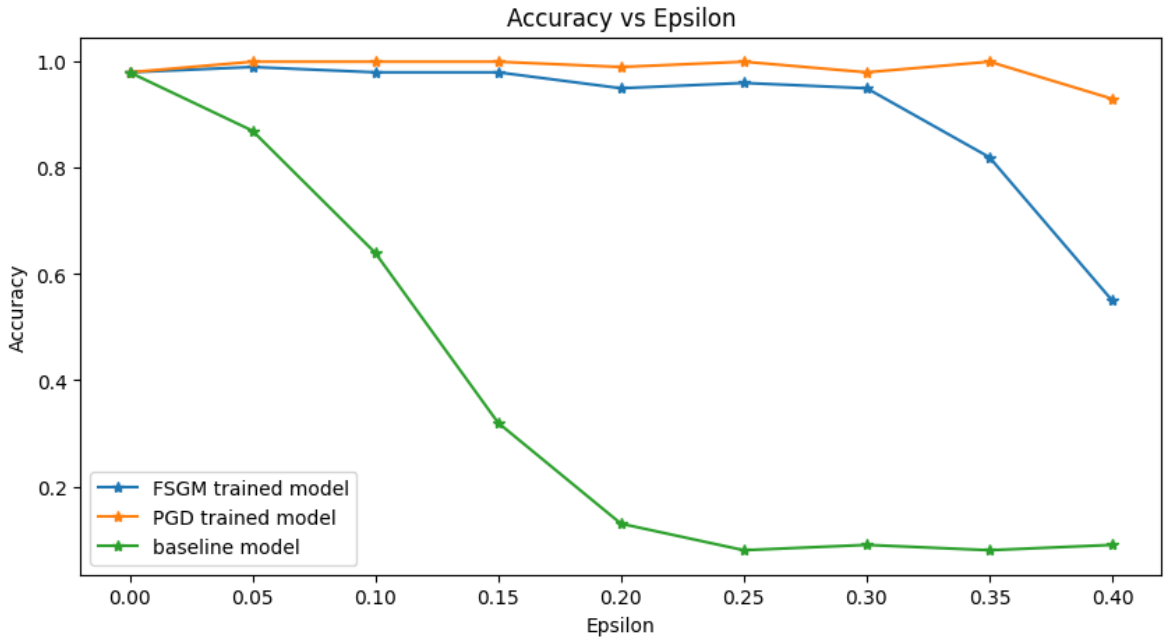

Figure 9: FGSM Adversarially Trained Model against PGD Adversarial Attack

Figure 9 displays the results of the PGD attack against the FGSM adversarially trained model compared to the baseline model. As expected, the PGD trained model had the highest robustness against the PGD attack. Moreover, the FGSM-trained model also showed a high level of robustness against the PGD attack compared to the baseline model. However, when the value of epsilon was increased, the accuracy performance of the FGSM-trained model eventually dropped.

3.3.2 PGD Adversarially Trained Model

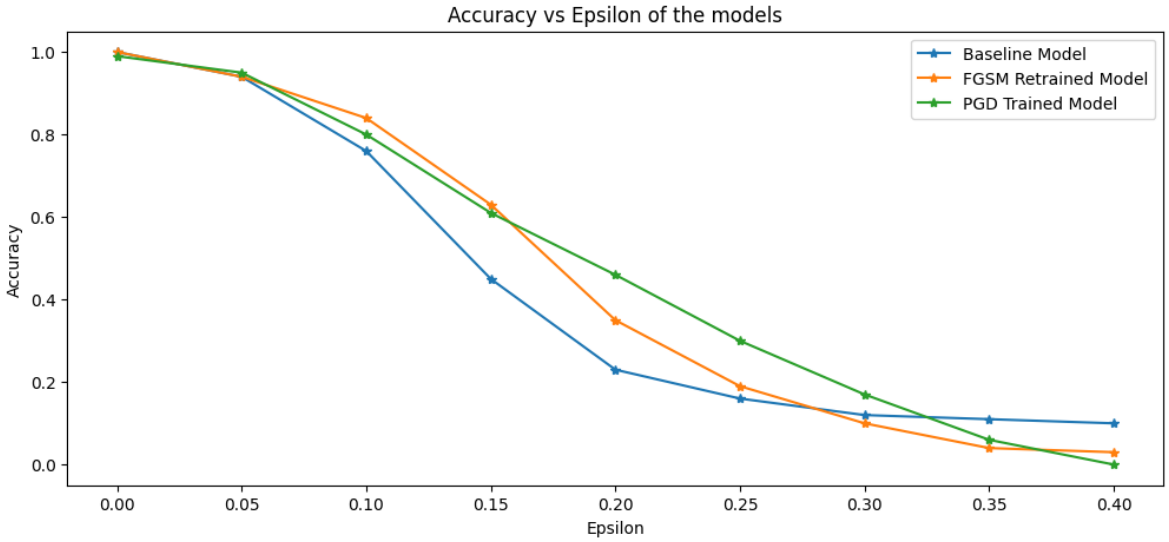

Figure 10: PGD Adversarially Trained Model against FGSM Adversarial Attack

Figure 10 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 showed better results as compared to the adversarially trained models. The PGD model accuracy surpassed the FGSM trained model around 0.15 epsilon, after which it generally provided a higher robustness as compared to the FGSM model.

Figure 11 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 predicted the class up to epsilon 0.15, the FGSM trained model accurately predicted the class up to epsilon 0.2, and the PGD trained model accurately predicted the class up to epsilon 0.25.

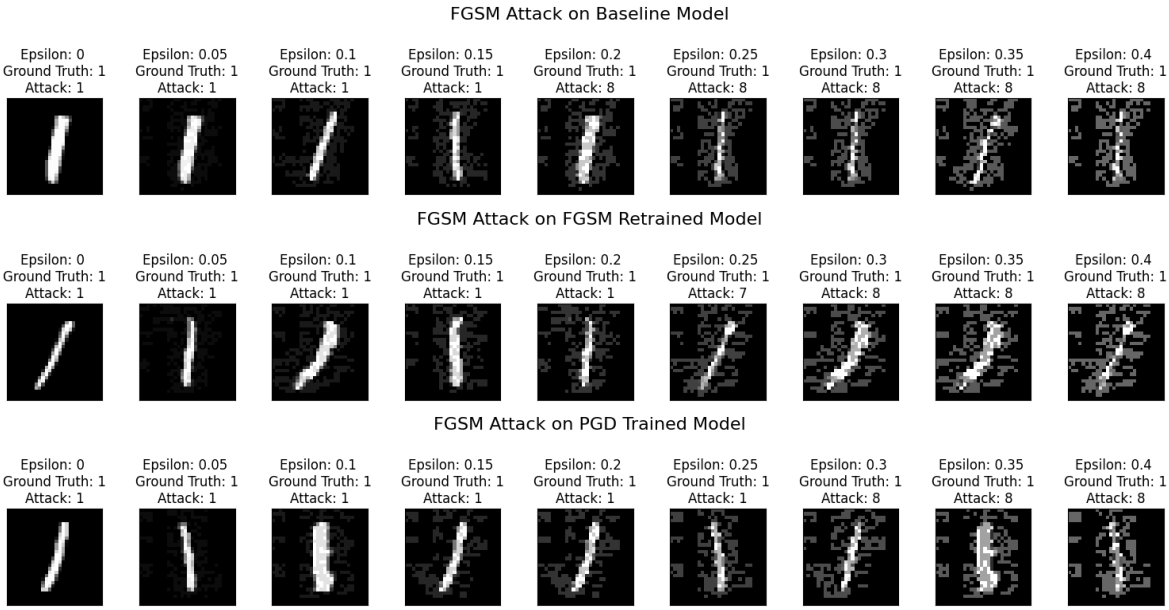

Figure 11: 10 Images showcasing the FGSM Adversarial Attack on the Baseline, FGSM Retrained and the PGD Trained Models

4 Analysis

Based on the results from the adversarial attack and training on the baseline CNN model, we can draw several conclusions about the relationship between attack parameters, adversarial training, and generalization of trained models.

4.1 Adversarial Examples

Our findings highlight the importance of the maximum range of perturbation (epsilon) in determining the effectiveness of both FGSM and PGD adversarial attacks. The relationship between epsilon and model accuracy is not linear, with a plateau in the decrease of accuracy after a certain epsilon value. For PGD attacks, we found that a small alpha value combined with a larger number of iterations was the most effective in generating adversarial samples for PGD [5]. This combination allows the attackers to infer the optimized perturbation that maximizes the model loss.

An intriguing observation was the misclassification of all perturbed images to a single class using FGSM on the baseline model, supporting Goodfellow et al.'s argument regarding the role of excessive linearity in the model [1].

4.2 Adversarial Training

We observed a slight decrease in accuracy for FGSM models when predicting non-perturbed images compared to the baseline model, confirming the robustness-accuracy trade-off discussed in the literature [9, 6]. Notably, the FGSM Retrained model outperformed both the untrained and baseline models, likely due to the retention of features from the benign dataset and the fine-tuning of the training model on the combined dataset, allowing for improved generalization and robustness [8].

For the two PGD adversarially trained models, their performance on benign samples are basically the same. Both models demonstrated a sudden decrease in accuracy when the epsilon value of the PGD attack passed a certain threshold, a phenomenon also reported in the original study [5]. The retrained model's faster decrease may be due to limited model capacity since it had already been trained with benign samples.

4.3 Generalization of Adversarially Trained Models

Adversarial training resulted in increased robustness against both PGD and FGSM attacks, which was due to the transferability property of adversarial examples [7]. As we had expected, we found high levels of robustness in the PGD trained models against PGD attacks and no decrease in accuracy for benign samples. The FGSM retrained model exhibited similar robustness to the PGD-trained model in both attack scenarios, possibly due to similar diversity of adversarial examples, hyperparameter choice, or the learning of robust features during training. Although the original PGD paper [5] suggested that PGD is the most effective first-order attack so far, the FGSM-trained model still seemed resistant to it in our experiments. The reason may be due to our choice of alpha when generating the poisoned data, which makes the perturbation for each image end faster than it should be, thereby decreasing the effectiveness of the attack [2].

Contrary to the findings of the original study, the increase in accuracy against FGSM for the PGD adversarially trained model compared to the baseline and FGSM trained model was not as high as anticipated [5]. This discrepancy could be attributed to our intentional decrease in the PGD model's performance, label leaking [3], overfitting, or the capacity of the model influencing the robustness of adversarial training [5]. Although the PGD model performed lower than expected against the FGSM transfer attacks, it still showed some resistance. In fact, for FGSM attacks with an epsilon value of 0.20 to 0.35, it outperformed the retrained FGSM model, which would generally be considered as the more robust model against FGSM attacks.

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] Tianjin Huang et al. *Bridging the Performance Gap between FGSM and PGD Adversarial Training*. 2022. arXiv: 2011.05157 [cs.CR].

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

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

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

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

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

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

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