Analyzing the Impact of Adversarial Examples on MNIST Dataset

Bennett Brain[#1], Chandler Smith[*2]

[#]*Department of Computer Science, Northeastern University*
360 Huntington Ave, Boston, MA, United States

[1]Brain.b@northeastern.edu

[3]smith.cha@northeastern.edu

Abstract

This paper investigates the effects of white-box and black-box adversarial attacks on a pre-trained convolutional neural network (CNN) for the MNIST digit classification task. The CNN architecture utilized has a proven success rate of 98% accuracy on the MNIST dataset without adversarial attacks. The study evaluates the efficacy of both targeted and randomly interspersed attacks on the training data, using various attack patterns.

The black-box attacks were carried out at training time by modifying certain subsets of the training data without any knowledge of the network's current weights. The white-box attacks included Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD) adversarial attacks with varying degrees of perturbation magnitudes.

The results reveal that both black-box and white-box attacks can significantly impact the model's accuracy, depending on the attack strength and the targeted digits. In some cases, the accuracy dropped to below random guesswork. Randomly distributed attacks, however, had little impact on the network's performance. The effectiveness of these adversarial attacks underscores the importance of understanding and addressing the vulnerabilities of CNNs to adversarial perturbations, highlighting the need for more research into robust defense mechanisms for deep learning models.

Keywords

Adversarial attacks, Convolutional neural networks, Fast Gradient Sign Method (FGSM), Projected Gradient Descent (PGD), White-box attacks, Black-box attacks

I. Introduction

Convolutional Neural Networks (CNNs) have demonstrated remarkable success in various image recognition tasks, including digit classification using the widely-studied MNIST dataset. While CNNs have achieved impressive performance in these applications, their vulnerability to adversarial attacks remains a significant concern. Adversarial attacks involve the introduction of carefully crafted perturbations to input data, which can cause the model to produce incorrect predictions, despite the perturbed data remaining visually indistinguishable from the original data to the human eye. Ensuring the robustness of CNNs against adversarial attacks is crucial for maintaining the security and reliability of systems reliant on these networks.

This study investigates the impact of two types of adversarial attacks on a pre-trained CNN designed for MNIST digit classification: white-box attacks on the accuracy of the network, and black-box attacks on the training process of the network. White-box attacks, specifically Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD), rely on knowledge of the model's architecture and parameters. In contrast, black-box attacks are performed at training time with no knowledge of the network's current weights. By examining the effects of these adversarial attacks on the model's performance, we aim to enhance the understanding of the vulnerabilities in CNNs and inform the development of effective countermeasures.

II. Related Work

One of the earliest studies investigating the vulnerability of neural networks to adversarial examples is the work by Szegedy et al. (2013). They discovered that neural networks are susceptible to adversarial perturbations, even when they achieve high generalization performance. Their

work demonstrates that adversarial examples are not solely caused by overfitting but are inherent properties of the networks, which could be exploited by adversaries.

Following Szegedy et al.'s work, Goodfellow et al. (2014) introduced the Fast Gradient Sign Method (FGSM), a widely studied white-box adversarial attack. The authors provided an explanation for the existence of adversarial examples and proposed a method for training more robust models using adversarial training. This technique involves augmenting the training data with adversarial examples to improve the model's resilience against such attacks.

Madry et al. (2017) presented the Projected Gradient Descent (PGD) attack, a more powerful white-box attack compared to FGSM. Their work focused on developing a framework for robust optimization, aiming to train deep learning models resistant to adversarial attacks. They demonstrated the effectiveness of this approach on MNIST and CIFAR-10 datasets, providing evidence that the proposed method can help enhance the robustness of convolutional neural networks.

The vast majority of research on adversarial attacks appears to focus on white-box attacks on already trained networks, but our black box attacks operate during training time. For that part of the project, we wanted to explore what could happen if an adversary had access to the training sets and could alter them, rather than attempting to fool a pre-trained network.

The field of adversarial attacks has established the vulnerability of neural networks and provided techniques to improve model robustness. Our research builds upon these foundational works to further explore the impact of white-box adversarial attacks, such as FGSM and PGD, on convolutional neural networks and investigate potential strategies to strengthen model defenses against these attacks.

III. Methods

As we focused on two types of adversarial attacks, we will be discussing them in two parts; the white box attacks on the accuracy of a pre-trained network, and black-box attacks on the training process of the same network structure. All of our experiments were done on the MNIST digits data set, using a Convolutional Neural Network with the following layers:

- A convolutional layer with 10 5x5 filters
- A max pooling layer with a 2x2 window and ReLu activation
- A convolutional layer with 20 5x5 filters
- A dropout layer with a 50% dropout rate
- A max pooling layer with a 2x2 window and ReLu
- A flattening operation into a fully connected layer with 50 nodes and a ReLu function
- A final fully connected layer with 10 nodes and log_softmax on the output

This network structure was chosen due to its proven success in accurately classifying MNIST. In the control group, when applied without any adversarial attacks, it achieves an accuracy of 98% on the data.

A. Black Box Attacks

Black Box attacks were carried out at training time. As training samples were being loaded in, they were modified in a particular way for each experiment, with no knowledge of the network's current weights used to inform the modification. Not every sample was modified, however; depending on the experiment, a certain subset of the training data was modified. The subsets were all samples for a given label, or selection of labels, though we also tried randomly interspersing the strongest adversarial attacks. Once the network trained on the manipulated data, we tested it on unmodified test sets to see how well it learned the actual features of the digits as opposed to the inserted attacks. Additionally, all training was done with the same fixed random seed in order to ensure consistency across tests.

Visualization of the attacks used can be seen below:

Fig. 1 The digit 8 modified by the first attack; two bright pixels in the top-left corner.

Fig. 2 The digit 8 modified by the second attack; 2x2 white boxes added to the top-left of imaginary 5x5 boxes at each corner.

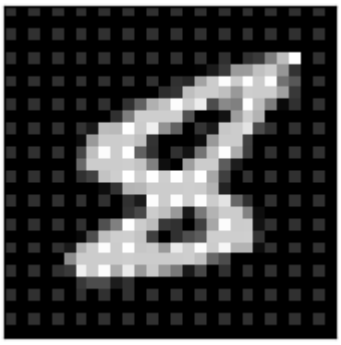

Fig. 3 The digit 8 modified by the third attack; a low-intensity pattern overlaid onto the image

Fig. 4 The digit 8 modified by the fourth attack; a diagonal line through the image

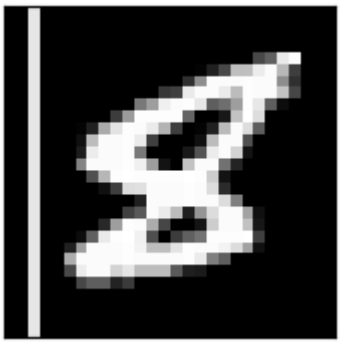

Fig. 5 The digit 8 modified by the fifth attack; a vertical line in the left-hand side of the image

The first attack was chosen because it would be a very light attack that should not cause too much confusion for the network. It was used as a lower bound on the attack efficacy. The second attack was chosen specifically because the network has 5x5 convolutional layers; thus each of the corners should see the same pattern, so we reasoned this may have a strong impact without visually changing the image itself too much. The third attack we assumed would be among the most invasive and cause the most damage to the network, but the underlying digit is still recognizable to human eyes.

The fourth and fifth attacks are similar, and are very invasive (especially the diagonal line). We hypothesized that the diagonal line would be the strongest attack and serve as an upper bound, and wanted to use the vertical line to see how effective a very strong signal in the data which would not intersect the digits themselves would perform compared to its diagonal sibling.

For the main results table, we ran solo attacks on all samples of the digit 8, and then group attacks on all samples in the set of 1, 3, and 7. Then, for further inquiry, we attempted using the strongest attacks randomly interspersed in the data at a frequency of 25% and 50%.

B. White Box Attacks

In contrast to black box attacks which are carried out without knowledge of the target model, white box attacks utilize full access to the model to generate adversarial examples. Two of the most well-known white box attacks are the Fast Gradient Sign Method (FGSM) and the Projected Gradient Descent (PGD) attack. These attacks are able to leverage gradient information from the target model

to systematically modify inputs in a way that fools the model into making incorrect predictions.

The FGSM attack works by calculating the gradient of the loss function with respect to the input, and then slightly perturbing the input in the direction that maximizes the loss. This single step of gradient ascent generates adversarial examples quickly, but the perturbations are often easily detected. The PGD attack builds upon FGSM by applying multiple smaller steps of gradient ascent while constraining the perturbation within a maximum norm bound. By repeating these projections over multiple iterations, the PGD attack is able to generate stronger adversarial examples that pose a greater threat to model robustness.

Visualization of the attacks used can be seen below:

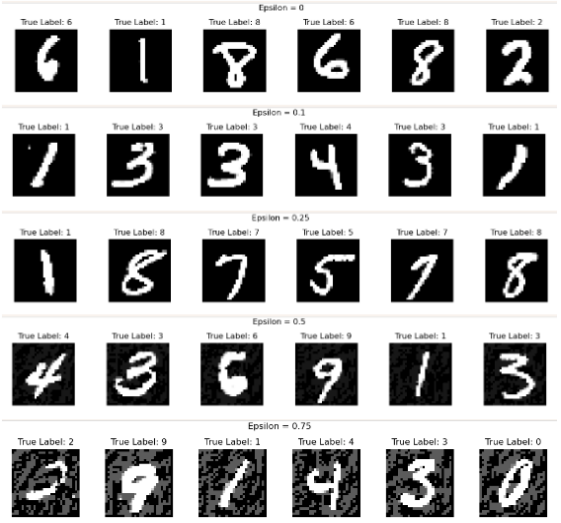

Fig. 6 The progression of FGSM attacks from epsilon=0 to epsilon=0.75.

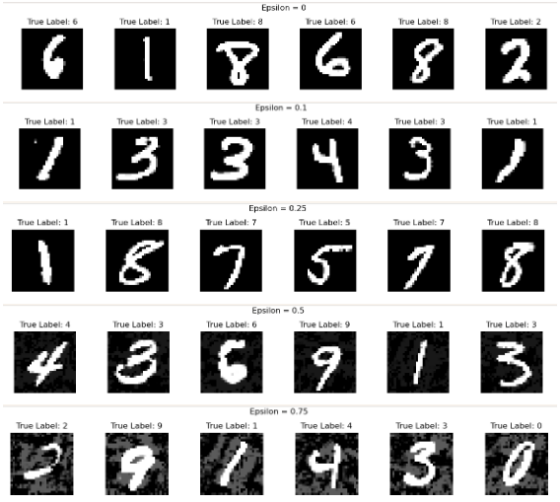

Fig. 7 The progression of PGD attacks from epsilon=0 to epsilon=0.75.

IV. Experiments/results

C. Black Box Attack Results

TABLE I
Single-Target Attacks

Attack Type	Measurement	
	Whole Test Set Accuracy	**Target Digit accuracy**
Corner Pixels	97.79%	97.33%
All-Corner Boxes	96.25%	78.64%
Pattern Overlay	97.59%	94.05%
Diagonal Line	91.10%	27.31%
Vertical Line	89.35%	8.73%
No Attack (Control)	97.89	97.64%

TABLE 2
Multi-Target Attacks

Attack Type	Measurement			
	Whole Test Set accuracy	**Target Digit accuracy (digit 1)**	**Target Digit accuracy (digit 3)**	**Target Digit accuracy (digit 7)**
Corner Pixels	97.89%	98.59%	98.42%	96.30%
All-Corner Boxes	94.86%	93.30%	83.96%	83.75%
Pattern Overlay	91.33%	65.02%	82.87%	82.30%
Diagonal Line	70.82%	0.53%	27.33%	10.80%
Vertical Line	72.92%	10.13%	21.29%	25.88%
No Attack (Control)	97.89%	98.33%	98.42%	96.30%

The effectiveness of the attacks can be easily understood from how low the accuracy became on the targeted digits, relative to the control group where no attacks were performed. Inserting the two corner pixels had almost no effect, showing that it is a fairly weak attack. The corner boxes did have some effect, causing a drop of about 10-20% accuracy depending on the shape of the underlying object. However, the effect was still weak relative to other attacks.

The diagonal line had severely pronounced effects, causing a drop to 30% accuracy or below (from a baseline of 96% accuracy or higher). In the case of the “1” digit in the multi-target attack, it dropped the accuracy to below random guesswork, all the way down to 0.53%. The variance in effect is likely due to the shape of the underlying samples,

but in all cases it had a very significant effect. The vertical line had a similarly strong effect, with its effectiveness also varying based on the shape of the underlying digit, sometimes even eclipsing the effect of the diagonal line. The fact that the diagonal line intersects the sample, while the vertical line often does not, has little effect on their outcomes. Rather, the shape of the sample is what matters.

The pattern overlay had odd results. When applied to a single digit, it had a pretty mild effect, dropping accuracy by only ~3.5%. When applied to a multi-target set, it had a much more pronounced effect, with the 1 digit dropping as low as 65%. In order to test whether this was due to the digit particulars or something intrinsic to targeting multiple samples, we ran another experiment with the same attack on 1 by itself and on the set (1,3,8).

TABLE 3
FURTHER EXPERIMENTS ON THE PATTERN OVERLAY

Attack Type	**Measurement**			
	Whole Test Set accuracy	**Target Digit accuracy (digit 1)**	**Target Digit accuracy (digit 3)**	**Target Digit accuracy (digit 8)**
Single Target	97.88%	98.50%	N/A	N/A
Multi Target	97.53%	97.80%	96.83%	93.22%

These results were confusing, to say the least. The most severely affected digit in the (1,3,7) set was completely unaffected when run alone, but the (1,3,8) set was nearly unaffected by the attack despite the significant effect on the (1,3,7) set. Due to time constraints, it was not possible to run thorough testing on just this one pattern, but such testing would be needed to draw any conclusions on the efficacy of this attack. What this data does indicate, however, is that there is a significant level of impact between samples in the data set- the weights learned from training on altered data for digits 3 and 7 affect how the network learns to handle digit 1.

Randomly distributed attacks were only tried on the most consistently effective attack types; the vertical and diagonal lines. Despite this, even high-frequency random attacks had nearly no impact on the network's accuracy, as seen in table 4. This is likely because randomly-distributed attacks have no correlation to the label IDs, and thus aren't helpful for the network to learn. If anything, the network would learn to ignore those signals in order to avoid confusion.

TABLE 4
RANDOMLY-INTERSPERSED ATTACKS

Attack Type and frequency	**Measurement**
	Whole Test Set Accuracy
Vertical Line, 25%	97.85%
Vertical Line, 50%	97.80%
Diagonal Line, 25%	97.76%
Diagonal Line, 50%	97.76%

In all of the targeted experiments, the accuracy of digits not in the targeted set was essentially unaffected by these attacks, as seen in Fig 8. As such, even though the network is learning how to understand specific adversarially-introduced patterns, that does not affect its ability to learn unaltered samples.

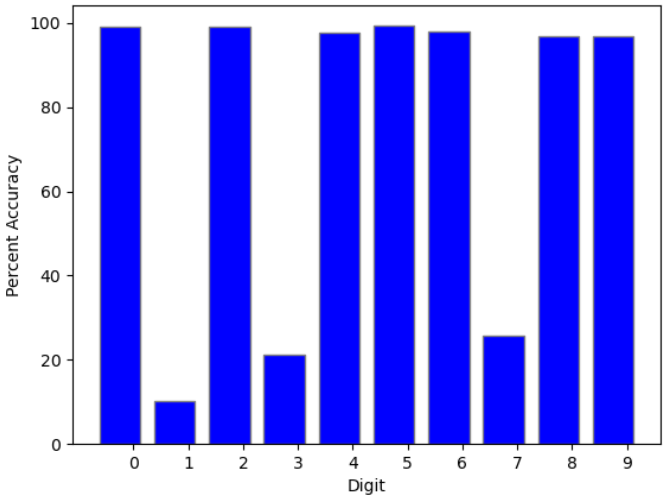

Fig. 8 Per-digit accuracy after the Vertical Line attack was run on the set (1,3,7)

In order to see how the network learned the poor data, we plotted the log-likelihood loss of training and test data during network training. As the attacks were done at training time, the loss there is measured against the adversarially altered data, which is why it decreases. The test loss gets worse with more epochs, as the network gets better at specifically learning the added adversarial signals. The graph in fig 9 was taken from the diagonal line attack on the (1,3,7) set, but similar patterns were observed across all effective attacks.

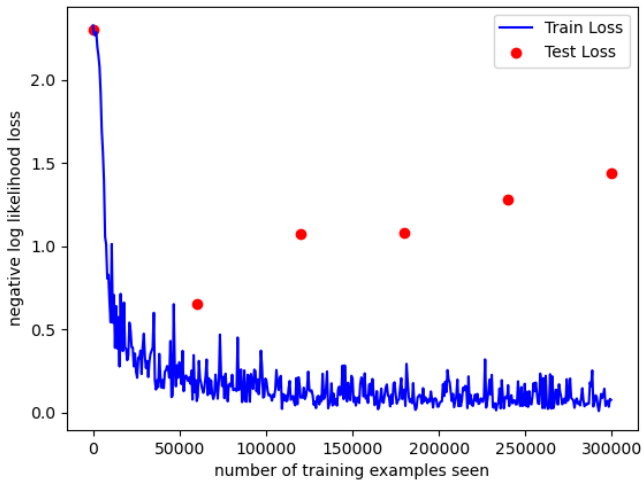

Fig. 9 negative log likelihood loss for the training and test sets while running the diagonal-line attack on the set (1,3,7)

To visualize what the network is learning, we looked at the weights of the first layer after a given iteration had finished training. The particular example in fig 10 was taken from the vertical-line attack applied to the (1,3,7) set. Filters 4, 5, and 7 exhibit strong verticality, indicating that those are the filters which learned the adversarial attack's structure. Filter 4 in particular nearly only learned a vertical line, which can be seen in the way it manipulated the sample data.

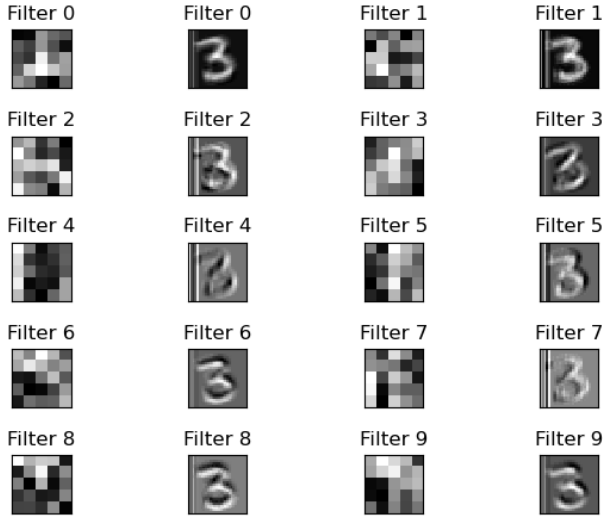

Fig. 10 The network's weights trained on the (1,3,7) set using a vertical line attack, applied to a post-attack sample

D. White Box Attack Results

TABLE 5
FGSM ATTACKS ON TEST SET

Epsilon Value	Average Loss	Accuracy
0	0.2439	98%
0.05	0.2556	97%
0.1	0.2681	97%
0.15	0.2815	97%
0.2	0.2958	96%
0.25	0.3114	96%
0.3	.03282	95%
0.5	0.5481	88%
0.75	1.4373	45%

From the given results, it can be observed that the FGSM attack has varying levels of effectiveness depending on the epsilon value. At lower epsilon values (0 to 0.15), the average loss increases slightly, but the overall accuracy of the model remains above 97%. This indicates that the attack is relatively weak at these levels, as the model is still able to maintain a high level of accuracy.

As the epsilon value is increased to 0.2 and 0.25, the average loss increases more rapidly, and the accuracy begins to drop (96%). This suggests that the attack is becoming more effective at these levels, causing a more pronounced impact on the model's performance.

At higher epsilon values, the effectiveness of the attack becomes even more apparent. For instance, at an epsilon value of 0.5, the average loss jumps to 0.5481, and the accuracy drops significantly to 88%. At the highest epsilon value of 0.75, the average loss skyrockets to 1.4373, and the model's accuracy plummets to 45%. This clearly demonstrates that the FGSM attack can be highly effective when the perturbations are of larger magnitude.

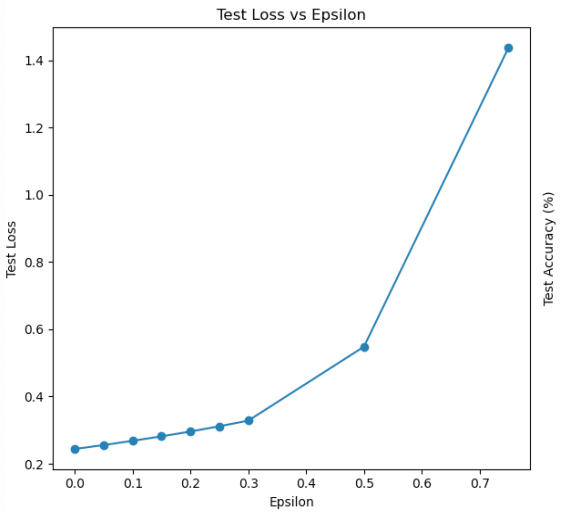

Fig. 11 negative log likelihood loss vs epsilon for the test set using the FGSM attack

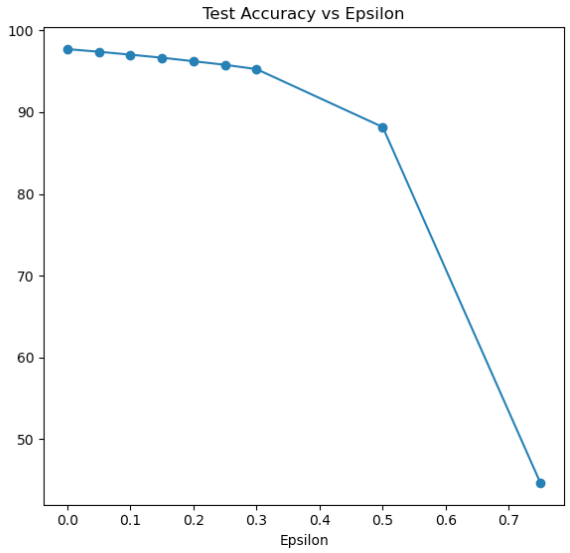

Fig. 12 test set accuracy versus epsilon using the FGSM attack

TABLE 6
PGD ATTACKS ON TEST SET

Epsilon Value	Average Loss	Accuracy
0	0.2439	98%
0.05	0.2581	97%
0.1	0.2734	97%
0.15	0.2895	97%
0.2	0.3069	96%
0.25	0.3257	96%
0.3	0.3457	95%
0.5	0.6446	86%
0.75	2.2700	13%

Looking at the results in table 6, it can be observed that the PGD attack demonstrates a similar pattern of effectiveness as the epsilon value increases when compared to the FGSM attack. For lower epsilon values (0 to 0.15), the average loss experiences a moderate increase, while the overall accuracy of the model remains stable at 97%. This implies that the attack has limited effectiveness at these levels, as the model's performance is not significantly impacted.

When the epsilon value is increased to 0.2 and 0.25, the average loss sees a more noticeable increase, and the accuracy starts to decline (96%). This suggests that the attack is becoming more effective at these levels, causing a more substantial impact on the model's performance.

At higher epsilon values, the PGD attack's effectiveness becomes even more evident. For instance, at an epsilon value of 0.5, the average loss rises to 0.6446, and the accuracy drops considerably to 86%. At the highest epsilon value of 0.75, the average loss surges to 2.2700, and the model's accuracy nosedives to 13%. This clearly demonstrates that the PGD attack can be extremely effective when the perturbations are of larger magnitude.

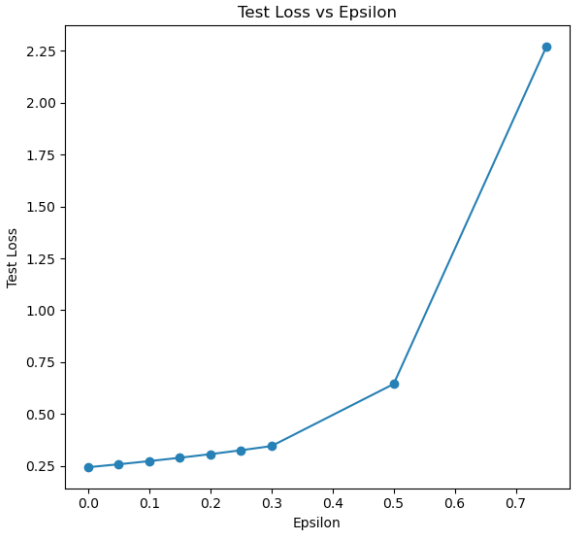

Fig. 13 negative log likelihood loss vs epsilon for the test set using the PGD attack

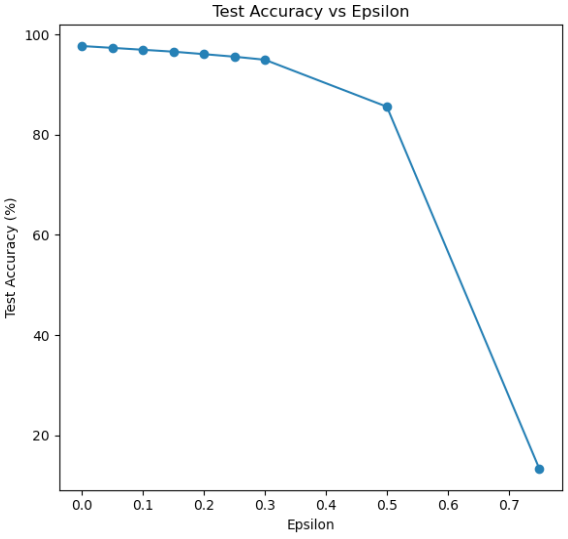

Fig. 14 test set accuracy vs epsilon using the PGD attack

V. Conclusions

Black Box adversarial attacks are difficult to perform in a way that is unobtrusive, but they are possible. Since no knowledge of the network is given, the attacks need to add powerful, consistent signals to the training data, especially in a convolutional neural network where a small collection of corner pixels will be flattened by the convolution, maxpool, and dropout layers. It is still possible to perform highly effective Black Box attacks which leave the underlying image easily classifiable to human eyes; the vertical and horizontal lines are examples of this, but the attacks would certainly be noticed by any human observing the training data. They cannot be stealthy attacks in the way that the white-box attacks are. Ultimately, the fact that a strong line which crosses the whole image is what it takes to trick a simple CNN into learning MNIST digit recognition poorly speaks to the robustness of convolutional neural networks for this task. And the absolute failure of randomly interspersed attacks to leave an impact also indicates that adversarial attacks at training time need to strongly correlate with a given label, or else they will be completely ignored by the network.

The effectiveness of strong signals such as the diagonal line in these attacks, however, does indicate the danger of training on data with undesirable signals. If most examples of a particular label in a dataset have a feature that one does not want the network to learn (for example, all pictures of dogs being on a grassy background), steps must be taken to avoid that feature dominating the network's learned weights.

Both FGSM and PGD white-box adversarial attacks demonstrate varying degrees of effectiveness depending on the magnitude of the perturbations introduced to the input data. For lower epsilon values, both attacks exhibit limited impact on the model's performance. However, as the epsilon value increases, the attacks become more effective, causing a more pronounced decline in the model's accuracy.

While these white-box attacks may not be as noticeable to the human eye as some black-box attacks, they still highlight the vulnerability of convolutional neural networks to carefully crafted adversarial perturbations. These attacks exploit the model's inherent sensitivity to input changes, enabling adversaries to manipulate the model's predictions while maintaining the appearance of a normal input to the human eye.

The robustness of convolutional neural networks against such white-box attacks is an essential area of research, given their widespread use in various applications. Ensuring the resilience of these networks against adversarial attacks requires a comprehensive understanding of their vulnerabilities and the development of effective countermeasures. Ultimately, the effectiveness of FGSM and PGD attacks on convolutional neural networks emphasizes the importance of exploring and implementing strategies to strengthen model defenses against adversarial attacks in the interest of security and reliability.

In conclusion, both black-box and white-box adversarial attacks pose significant challenges to the security and reliability of convolutional neural networks. While black-box attacks may be more noticeable to human observers, white-box attacks such as FGSM and PGD can be stealthy and highly effective under certain conditions. Therefore, it is crucial to invest in ongoing research to better understand these vulnerabilities and develop robust defense mechanisms to protect against adversarial attacks in various applications.

Acknowledgment

We wish to acknowledge professor Bruce Maxwell for teaching the Computer Vision class this project was designed for, as well as providing helpful lecture notes and papers to reference.

References

[1] Koehlor, G. (2020, February 17). *MNIST handwritten digit recognition in pytorch*. Nextjournal. Retrieved April 4, 2023, from https://nextjournal.com/gkoehler/pytorch-mnist

[2] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks.

[3] Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples.

[4] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks.

[5] Zhang, O. (2022, July 21). *Intro_to_ml_safety/main.md at master · centerforaisafety/intro_to_ml_safety*. GitHub. Retrieved April 24, 2023, from https://github.com/centerforaisafety/Intro_to_ML_Safety/blob/master/Adversarial%20Robustness/main.md