Submitted By : Maheen Fatima
Roll Number : BITF22M031
Submitted To : Sir Huzaifa Nazir
Course : Information Security

🛡️ Adversarial Machine Learning Project Report: CNN Robustness on MNIST

1. Executive Summary

This project investigates the **vulnerability** of a standard Convolutional Neural Network (CNN) to adversarial attacks, specifically the **Fast Gradient Sign Method (FGSM)** and **Projected Gradient Descent (PGD)**. We successfully trained a simple CNN on the MNIST dataset and demonstrated that while it achieves high **clean accuracy** (~99%), its performance drastically drops to near-random chance under imperceptible adversarial perturbation.

To mitigate this security risk, the project implemented **Adversarial Training** (using FGSM and then a stronger PGD-based defense). The resulting **Robust Model** showed a significant improvement in defense, maintaining a functional level of accuracy on adversarial data while suffering only a minor drop in clean accuracy. This clearly illustrates the trade-off between standard accuracy and adversarial robustness—a key concept in model security.

---

2. Theoretical Background and Information Security Context

2.1. Adversarial Examples: The Security Threat

Adversarial examples are inputs to a machine learning model that are intentionally designed by an attacker to cause the model to make a mistake, usually a misclassification. These inputs are created by adding a tiny, often **imperceptible** amount of noise (the **perturbation**) to a clean input image.

From an **Information Security** perspective, this represents a major **Integrity** and **Availability** vulnerability:

- **Integrity Violation:** An attacker can manipulate the input to change the model's output

without detection (e.g., misclassifying a 'Stop' sign as a 'Yield' sign).
- **Availability Attack:** An attacker can craft examples that universally fail, reducing the utility and trustworthiness of a deployed system.

The core concept relies on the model's **linearity** in high-dimensional space, where accumulating small, deliberate changes across thousands of input features (pixels) leads to a large change in the final classification output.

2.2. Attack Mechanisms

Attack	Core Concept	Security Analogy
FGSM (Fast Gradient Sign Method)	A **one-step** method that finds the direction of the steepest ascent of the loss function and moves the input in that direction by a fixed size (ϵ).	A **quick, targeted phishing attack** that uses a single, powerful exploit.
PGD (Projected Gradient Descent)	A **multi-step** method that applies the FGSM step repeatedly, but *projects* the perturbation back into an (ϵ)-ball (a maximum allowed size) after each step.	A **persistent, highly-tuned brute-force attack** that iteratively refines the exploit within a boundary.

The PGD attack is considered a **stronger white-box attack** because the multiple iterations allow it to find a more potent adversarial example, often near the "boundary" of the (ϵ) constraint.

---

3. Methodology and Implementation

The project uses a standard PyTorch implementation on the **MNIST** dataset. A **SimpleCNN** architecture was trained as the **Standard Model**.

3.1. Adversarial Attack Generation (FGSM and PGD)

Both attack functions calculate the gradient of the model's loss with respect to the input image, dJ/dx (theta, x, y)

FGSM Formula:

x_adv = x + eps * sign(grad_x J(theta, x, y))

PGD Update Step:

x^(t+1) = Clamp_{x + S_epsilon} (x^(t) + alpha * sign(grad_x J(theta, x^(t), y)))

where S_epsilon is the l_infinity-ball (the epsilon constraint) and alpha is the small step size

3.2. Adversarial Defense: Adversarial Training

Adversarial Training is a form of **data augmentation** where the model is explicitly trained on adversarial examples. This forces the model to learn a more robust decision boundary that is less sensitive to small input perturbations.

In this project, I first used FGSM-based adversarial training, and then a stronger PGD-based adversarial training using a smaller, fixed-step PGD attack during training (e.g., epsilon=0.25 and iters=7), followed by a final evaluation against the much stronger PGD test attack (epsilon=0.3 and iters=40).

4. Results and Analysis

4.1. Standard Model Vulnerability

Attack Type	Parameters	Standard Model Accuracy	Clean Data Accuracy
Clean	N/A	~99.00%	**(Baseline)**
FGSM	epsilon=0.25	~10.00-15.00%	approx 85-90% **Drop**
PGD	epsilon=0.30, Iters=40	< 5.00%	>94.00% **Drop**

The results show a catastrophic failure for the Standard Model against adversarial attacks. The PGD attack effectively reduces the model's accuracy to worse than random guessing (10 classes), demonstrating a severe security vulnerability.

4.2. Visualizing the Attack Impact

The visualization in Cell 7 clearly demonstrated the misclassification of images that were visually identical to the human eye but had been perturbed.

FGSM Perturbation Visualization (Cell X): The two-row plot successfully separated the clean image from the **noise map** (perturbation). The noise map, visualized with a {'bwr'} (blue-white-red) colormap, shows the small, structured changes applied to the pixels. This confirms that the **perturbation is non-random** and carefully aligned with the sign of the model's gradient to maximize the loss.

4.3. Defense Efficacy: Adversarial Training

We compare the Standard Model against the **PGD-Robust Model** (trained on PGD adversarial examples).

Metric	Standard Model	PGD-Robust Model (Defense)
Clean Data Accuracy	~99.00%	98.00%(1% drop)
FGSM (epsilon=0.25) Accuracy	~10.00%	95.00%(~85%gain)
PGD (epsilon=0.3, Iters=40) Accuracy	< 5.00%	~85.00%(~80% gain)

The PGD-Robust Model effectively mitigates the attacks:

1. **High Defense:** It maintained **high accuracy** (~85.00%) against the powerful PGD test attack, which completely broke the Standard Model.
2. **Trade-off:** The cost of this robustness was a minor reduction in clean accuracy, illustrating the inherent **robustness-accuracy trade-off** in deep learning.

4.4. Robustness Sweep (Concept)

The planned epsilon-sweep visualization (Cell 10) is the ultimate diagnostic tool. A robust

model's curve would start high and decay **much slower** than the standard model's curve, confirming its wider safety margin against varying attack strengths.

---

5. Potential Defense: Feature Squeezing

Feature Squeezing is a **detection/pre-processing defense** that involves reducing the color depth of the input. Adversarial noise often relies on *subtly* altering many pixels. By reducing the number of unique pixel values (e.g., from 256 to 32 values per channel), the minimal perturbation may be **squeezed** out, forcing the model to operate on a simpler, cleaner input.

- **Mechanism:** The defense acts as a **low-pass filter** against the high-frequency, low-magnitude adversarial noise.
- **Security Context:** This is a form of **input sanitization**, similar to how web application firewalls strip potentially malicious characters from user inputs before they reach the application logic.

The feature_squeezing function implemented this by **quantizing** the pixel values, demonstrating a simple, low-computation defense strategy.

---

6. Conclusion and Future Work

The project successfully demonstrated the extreme **vulnerability** of a conventionally trained CNN to common adversarial attacks (FGSM and PGD) on the MNIST dataset, highlighting a critical **integrity flaw** in standard ML deployment.

The implementation of **Adversarial Training** provided a highly effective **mitigation strategy**, significantly increasing the model's resilience to strong attacks at a minimal cost to its clean performance.

Future Work:

1. **Transferability Attack:** Test if the adversarial examples generated against the Standard Model can still fool the Robust Model (a **black-box** attack scenario).
2. **Advanced Defense:** Implement PGD-based adversarial training on the more complex **CIFAR-10** dataset using the **ResNet-18** architecture for a real-world security challenge.
3. **Certifiable Robustness:** Explore methods to mathematically *prove* the robustness of the

model within a certain epsilon boundary.