

Project Title: Adversarial Security Audit: CNN Robustness on MNIST

1. Project Idea and Background

Core Idea: Demonstrate and mitigate the security flaw of Deep Learning models caused by **Adversarial Examples** (invisible input noise), which is a critical Integrity failure.

The Hook: Adversarial noise is the AI equivalent of getting photo-bombed. A perfect input is ruined by a tiny, targeted disturbance.

2. Methodology and Key Results (The Trade-Off)

A Simple CNN was trained on MNIST, attacked by FGSM and the strong PGD attack, and defended it with PGD-based Adversarial Training.

Standard Model Audit (Vulnerable):

- **Clean Accuracy:** 99.00 percent.
- **Vulnerability:** Under the PGD attack, accuracy fell to **less than 5.00 percent**.

PGD-Robust Model (Defense Success):

- **Robustness Gain:** The model gained 80.00 percentage points in accuracy against the PGD attack.
- **Clean Accuracy Cost:** This robustness required a minimal trade-off, losing only 1.00 percent of clean accuracy.
- **Defense Result:** Final model accuracy against PGD was 85.00 percent.

3. Accomplishments and Future Steps

What Has Been Done and Working:

- **Attack Auditing (MNIST):** Working PyTorch functions for FGSM and PGD attacks are complete.
- **Defense Implemented (MNIST):** Successfully trained a model using PGD-based Adversarial Training.
- **Visualization:** Generated visual proof of the attack, mapping the structure of the perturbation noise.
- **Next-Step Architecture: Code is implemented and ready** for scaling the project to **CIFAR-10** using the **ResNet-18** model.
- **Defense Ready:** Implemented **Feature Squeezing** as an alternative input sanitization defense.

Conclusion:

The project proves that standard CNNs are fragile, but Adversarial Training is a highly effective security mitigation, transforming a vulnerable model into a robust defender at minimal cost.

Future Work:

Execute the established code blocks to fully train and audit the ResNet-18 model on CIFAR-10, then investigate Transferability Attacks.