Structure and Properties of Adversarial Noise

Sharon Qian, Irina Tolkova
sharonqian@g.harvard.edu, itolkova@g.harvard.edu

HARVARD
John A. Paulson
School of Engineering
and Applied Sciences

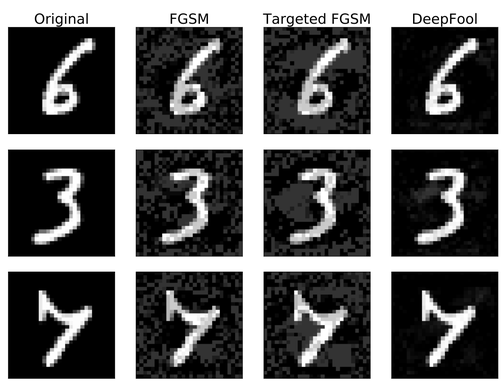
Introduction

In 2013, Szegedy et al. discovered that neural networks could be effectively targeted with adversarial noise: very small perturbations to inputs that results in high-confidence misclassification. Since then, adversarial examples have been widely studied, with many suggested attacks and counterattacks. For this project, we:

- Compare the properties of adversarial and random noise.
- Show that we can detect adversarial samples with high accuracy.
- Present some extensions of the fast gradient sign algorithm to construct images that appear to be white noise, but which are misclassified with high confidence.

What is Adversarial Noise?

Adversarial noise is a small perturbation of clean data which can fool a classifier into predicting an incorrect label. There are numerous methods to generate these perturbations, and we implement three such methods: Fast Gradient Sign, Targeted Fast Gradient Sign, and DeepFool. The figure at right shows examples of original images along with generated adversarial images (misclassified with high confidence).



Consider a classifier f, and let θ be the parameters of the classifier, x be the input, y be the targets associated with x, and J be the loss used to train the classifier. The three algorithms are detailed below:

Fast Gradient Sign (FGS): This algorithm, developed by Goodfellow et al. in 2014, relies on moving down the gradient of the loss with respect to the class label until a classification boundary is crossed. Each step is updated as:

$$\eta = \epsilon \text{sign}(\nabla_x J(\theta, x, y))$$

Targeted FGS: This variant of FGS iteratively moves up the gradient of the loss J with respect to the target label y_{target} corresponding with the desired misclassification class:

$$\eta = -\epsilon \text{sign}(\nabla_x J(\theta, x, y_{target}))$$

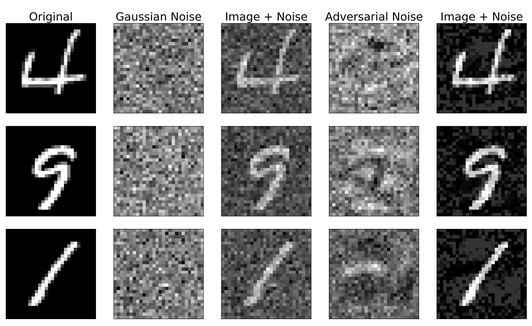
Deep Fool: DeepFool improves upon Goodfellow's algorithm, and constructs adversarial examples with less visible noise. The method consists of iteratively descending down the gradient of the classifier, scaled by the classification confidence:

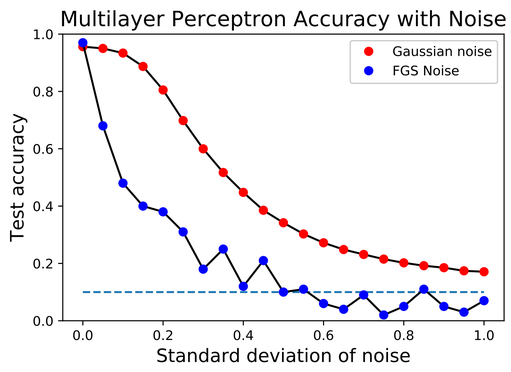
$$\eta = -f(x)\frac{\nabla f(x)}{||\nabla f(x)||_2^2}$$

How Effective is an Adversarial Attack?

The main property which makes adversarial noise malicious is that the perturbation can be very small in magnitude, and seem imperceptible to a human eye. To see how it compares to random (Gaussian) noise of the same magnitude, we:

- Trained a multilayer perceptron (MLP) on the MNIST dataset.
- Implemented one-step FGSM with the given step size to construct a small adversarial test dataset for a given noise budget.
- Calculated the test accuracy of the MLP.
- Repeated this experiment with Gaussian noise of equivalent magnitude.





The left figure shows examples of images and noise; the right figure shows classification accuracy across noise budgets. Even **simple adversarial algorithms result in much higher misclassification rates at a much lower magnitude than random noise**.

Can Adversarial Attacks be Detected?

If a classifier is subjected to an adversarial attack, one possible defense may be to design a (binary) classifier to detect adversarial samples. For this approach, we:

- Trained a multilayer perceptron (MLP) on the MNIST dataset.
- Generated images corrupted by adversarial noise from three different algorithms against this classifier.
- Trained a multilayer perceptron (MLP) to distinguish between clean original images, and noisy adversarial images.
- Measured its detection accuracy, and compared this to the accuracy of detecting images corrupted by Gaussian noise of the same noise budget.

Adversarial Attack	μ_{noise}	σ^2_{noise}	Detection Rate of Adversarial	Detection Rate of Gaussian
FGS	0.0102	0.0531	100%	99.9%
Targeted FGS	0.00416	0.0622	100%	80%
Deep Fool	0.00207	0.00313	99.0%	57.5%

By design, the three methods of generating adversarial attacks only add a small magnitude of noise relative to the original image. Yet surprisingly, **these corrupted images can be detected by an MLP with high confidence**, unlike images with Gaussian noise of the same magnitude.

Algorithmic Extensions

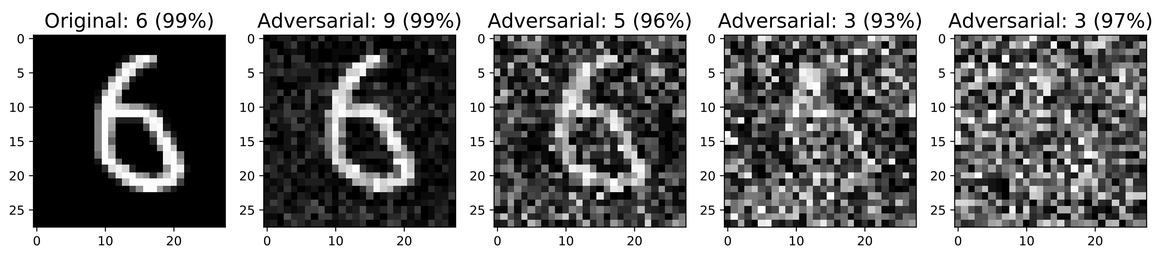
If adversarial noise can be detected, can it be reversed? In particular, what kind of image will be produced by repeated applications of the Fast Gradient Sign Method?

Algorithm 1: k-Iterative Attack

for i = 1 **to** k **do**
  for j = 1 **to** steps **do**
    $x_i \leftarrow x_i + \epsilon \cdot FGSM(x_i, y_i)$
  end for
  Check $y_i \neq f(x_i)$
  $y_i \leftarrow f(x_i)$
end for

To explore these questions, we use the algorithm described on the left. We repeatedly calculate adversarial steps, and update the reference image after a fixed number of iterations. We used $\epsilon = 0.04$, steps $= 5$, and $k = 500$.

The figure below shows the resulting images, class predictions, and confidence values for an example original image and subsequent images every 10 iterations.



While our initial motivation was to attempt to reverse an adversarial attack, the algorithm does not return to the original image. Instead, it moves through different classes with high confidence, while the image becomes increasingly indistinguishable.

For a collection of original images, we ran the algorithm for $k = 500$ iterations and tested the statistical properties of the resulting image. By the Ljung-Box test, the output was not statistically distinguishable from white noise. We conclude that **this algorithm results in a method for generating white-noise-like images that are misclassified with high probability.**

Future Work

In continuation of this project, we would like to:

- **Formalize the ability to detect adversarial images in a PAC-learning framework**, by comparing the sample complexity of the binary detection problem to the sample complexity of the original classifier.
- Use the k-iterative variant of FGS to **understand decision boundaries around a given image**. In particular, the algorithm seems to exhibit cyclic behavior if we switch the direction of the gradient as soon as the boundary is crossed.
- The statistical similarity to white noise may be difficult to evaluate on MNIST data, which has limited variation in image structure. For future work, we would like to **extend experiments to more complex datasets** such as ImageNet.