**Deep Learning Final Project**

**Adversarial Attacks**

1. **Introduction**:
2. **What Are Adversarial Attacks:**

Deep neural networks have revolutionized the task of computer object recognition across many fields. Since around 2013, computers perform as well as humans or even better at several tasks including facial recognition, image recognition and more. These deep learning algorithms perform very well and are generally thought to be not susceptible to outliers and biases in data, but they are not perfect and there are ways to “fool” even some of the most accurate recognition networks known today.

Adversarial attacks are a specific technique to fool a neural network. *These are algorithms which create adversarial examples*, which are input samples to a network that have been modified in a way that the target network no longer classifies them correctly. These modifications are not random, are carefully computed and are called “perturbations”. The task which adversarial attacks aim to optimize is the finding of specific points in an input sample which are most influential in a network’s classification decision, and then changing these specific points.

1. **Why Are They A Problem:**

Once an algorithm creates effective adversarial samples, it can be used for malicious purposes. A deep learning model that recognizes malware, for example, can be fooled into classifying dangerous software as safe. The common use of deep learning models for computer classification tasks makes adversarial attacks a real-world problem.

1. **White Box Attack:**

White box adversarial attacks are algorithms constructed in a best-case scenario (for the attacker) where the attacker knows the target model and all its parameters. With this knowledge at hand, the white box algorithm uses the target model itself to create adversarial samples and test their accurate classification rate.

1. **Black Box Attack:**

A scenario where the attacker does not know the parameters of the target network or does not even know the model or algorithm the target is using. The attacker can only use the target algorithm as a “black box” which outputs a classification per input but cannot actually create the adversarial samples through an optimization on the target model’s parameters. An algorithm in a black box scenario must find some other model or strategy to create its adversarial samples, which we will explore in this project.

1. **Our Project Goals:**

* Research white box attacks and their effectivity
* Research black box attacks and their effectivity
* Conclude and generalize over some optimal strategy for creating adversary samples / attacks
* Study a way to fight adversarial attacks and limit their damage to a deep learning model

1. **Project Overview:**
2. **Step one: White box attacks:**

We first trained two separate networks, one on the MINST dataset and the other on CIFAR10. We then create adversarial samples using a fast gradient sign method (FGSM) described below. We run the trained networks on the formulated adversarial samples and record the results as follows.

1. **Step two: Black box attacks:**
2. **Step three: Creating an “adversary-protected model”:**
3. **Step one: White Box Attacks**
   1. **General:**

We used two neural networks:

* + ResNet:
  + SqueezeNet:
  1. **Results and accuracies:**
  2. **Creating Adversarial Samples Using FGSM:**

We use the gradient of the loss function with respect to the input image in order to find a perturbation which we will add to our original images:

X\_adversarial = X\_original + ε \* GRADx(J(θ, X\_original, y))

X\_original = the original input samples

GRADx = the gradient with respect to the input image

J = the model loss function

ε = our perturbation addition factor

y = the original target value

θ = the model weights