CompSci 175: Project in Artificial Intelligence

(Adversarial Machine Learning)

Attack Project Report

by

Team Headbutt

Jaime Markkern (jmarkker)

Kenny Tran (kennyt11)

Junyan Wu (junyanw3)

The approach we utilized was the Iterative Fast Gradient Sign Method (IFGSM) also known as Basic Iterative Method (BIM). IFGSM is a white-box method where the attack generates adversarial examples by iteratively applying the Fast Gradient Sign Method (FGSM) attack multiple times.

FGSM is an attack algorithm that uses the gradient of the loss function for predictions to make a machine learning model to predict the wrong class or a targeted class of our choice. FGSM takes in the loss function and calculates the gradient with respect to the input image instead of other parameters (with respect to x not theta). Since we are using a targeted attack, we calculate the gradient of the loss function for our targeted image and try to change the input image slightly towards the target. If this were not a targeted attack, we would calculate the gradient of the loss function for the original prediction and try to move along this gradient, increasing the chance of making an incorrect prediction. We can use this gradient function to obtain an adversal image by calculating the sign of the gradient and multiplying it by a value (epsilon or B), and then adding it to our original image. If we apply a small epsilon value, this would minimally warp the original image that the human eye would not notice. However, for a computer, it will be confused by this change and make a wrong prediction.

$$\text{targeted attack: } x' = x - B\,sign\,(\nabla x\, L(x, y_t))$$

$$\text{untargeted attack: } x' = x + B\,sign\,(\nabla x\, L(x, y_p))$$

We originally used FGSM without iteration based on the example provided and realized that although the runtime was short, the distance was too great, so we decided to use the IFGSM method to solve this attack problem by improving our score and distance, and sacrificing the runtime.

We ran FGSM repeatedly with a for loop a number of times (steps). By testing out different values of epsilon as shown in *Figure 1: Epsilon and Score*, we found that an epsilon value of *0.0055* generated the best score with a score of

95.88548065185546. We also experimented by varying the number of iterations of FGSM and found that iterating with a step size of 10 generated the best score as shown in *Figure 2: Step Size and Score*.

In conclusion, we feel confident with the results submitted because we experimented with different variations of epsilon and step size values to achieve the most “optimal” score using the IFGSM algorithm.

Epsilon	Score
0.001	26.21828851064046
0.005	94.71878304375542
0.00575	95.86171508789062
**** 0.0055 ****	**** 95.88548065185546 ****
0.006	95.83749462127685
0.007	95.7327554321289
0.008	95.61984071095783
0.009	95.50500878651935
0.01	95.38556966145832

Figure 1: Epsilon and Score

Step Size (using Epsilon = 0.0055)	Score
5	80.40618824852837
**** 10 ****	**** 95.88548065185546 ****
15	95.45501501719157
20	94.99588924407959

Figure 2: Step size and Score