Adversarial Machine Learning with Untargeted Fast Gradient Sign Attack on CIFAR10 EfficientNet

Sean Johnson
University of Massachusetts
Amherst
Amherst, MA 01003, USA
seanjohnson@umass.edu

***Abstract*—Generating adversarial images to an image classifier can require nontrivial computational power. This work shows that a fast gradient sign attack on a convolutional neural network provides an efficient way to generate a set of adversarial images large enough to train a new network on. In this project, I evaluate the effectiveness of this attack and possible defenses. My findings show that this attack is effective, but less so when used on a different network than the one used to generate the adversarial images. My findings also show that adversarial training and gaussian filtering can be used as a defense.**

I. Introduction

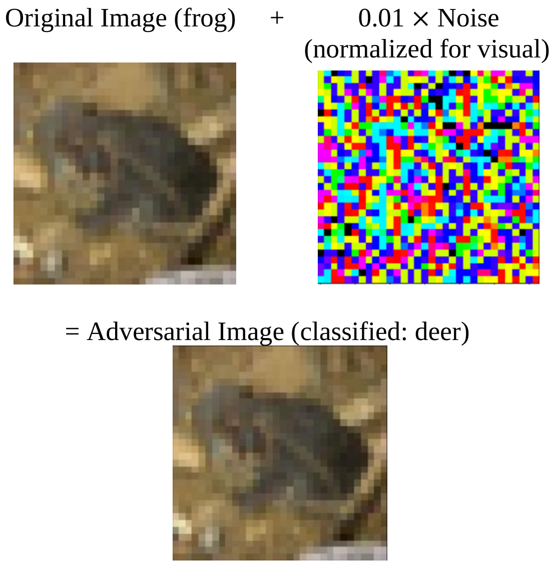
Machine learning (ML) is a tool for data analysis and classification that has grown in importance in the past decade as its use has expanded into areas such as self-driving cars, medical diagnoses, and speech recognition. One challenge that is present in many of these applications is image classification, for which many ML algorithms have been developed. Of these, convolutional neural networks (CNN) are a popular choice. The security of these networks has been a point of great concern as these algorithms find use in safety-critical applications. Szegedy et al. [4] first discovered the susceptibility of deep neural networks (DNN) to misclassify images modified by an adversary. Through noise injection attacks, the input of a DNN can be optimized to maximize the prediction error [2, 4]. This paper evaluates the effectiveness of an untargeted fast gradient sign (FGS) attack on a CIFAR10 EfficientNet. It also evaluates the effectiveness of adversarial training and preprocessing to defend against it. My contributions are as follows:

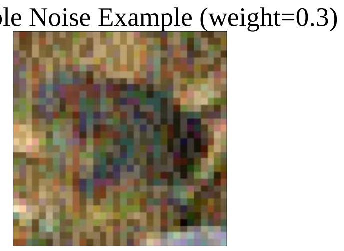
- I evaluated the attack effectiveness on a simple deep CNN to see if the attack translates to different architectures.
- I trained the simple deep CNN on the adversarial training set and evaluated its effectiveness against adversarial images.
- I applied a gaussian filter as pre-processing to evaluate its effectiveness against adversarial images.

These experiments provide a glimpse at the attack and defense possibilities of artificial neural networks. Adversarial ML is an important field of research to ensure our safety in the future.

II. Untargeted Fast Gradient Sign Attack

First developed by Szegedy et al [4], FGS attack is a noise injection attack for neural networks that do classification. The attack can be targeted or untargeted, where an untargeted attack minimizes the confidence of the original label and a targeted attack maximizes the confidence of a target label. The attack generates adversarial noise by calculating an "imperceptibly small vector whose elements are equal to the sign of the elements of the gradient of the cost function with respect to the input" [1]. This noise gets added to the original image, which causes it to be misclassified. If adding noise once does not cause the image to misclassify, then more iterations of adversarial noise are added to the image. In this project, the noise was multiplied by 0.01 before being added to the original image. This weight was chosen because higher weights resulted in perceptible noise in the adversarial image.





FGS attack can be computed efficiently because it uses a gradient. Other attacks like the "one pixel attack" [3], which uses differential evolution, are more computationally expensive.

Average Attack Δt	
FGS	***One Pixel***
0.089s	12.2s

One goal of this project was to generate an adversarial counterpart to the CIFAR10 image set, which contains 60,000 labelled images. All network training and adversarial image generation for this project was done on an Amazon Web

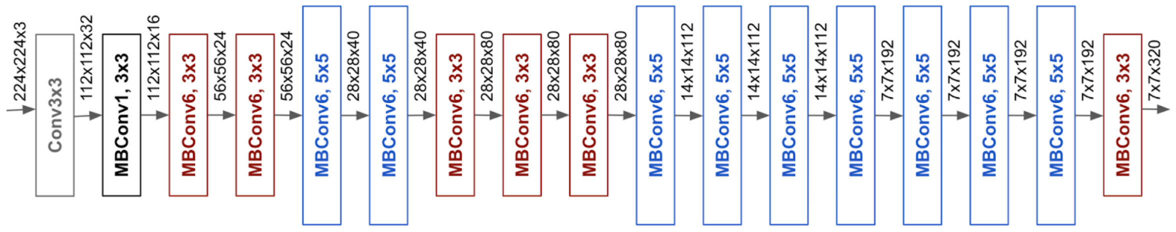


Fig. 1. EfficientNet Architecture

Services EC2 p2.xlarge instance. With that being the available computational power, the average time for a one pixel attack was orders of magnitudes larger than the time for a FGS attack. Therefore, the FGS attack was chosen as the focus for this project. With an untargeted FGS attack on a CIFAR10 EfficientNet, it was possible to generate adversarial counterparts for all images in 1.5 hours. All images were successfully perturbed to be misclassified.

III. CIFAR10 Image Set, EfficientNet, Simple Deep CNN

It worthwhile to briefly explain the tools used during this project. The CIFAR10 image set is split up into a training set of 50,000 labelled images and a test set of 10,000 labelled images, where the training set is intended for model training and the test set is intended for model verification. Any reference to "test set" in this paper is referring to this test set. EfficientNet (Fig. 1.) [5] is a deep CNN made by Google that was made to be resource efficient. Lastly, "simple deep CNN" (Fig. 2.) is self-explanatory, although it is notably less deep than EfficientNet. It is provided by the developers of Keras.

IV. Adversarial Test on Simple Deep CNN

After implementing the untargeted FGS attack on the CIFAR10 EfficientNet, the next step in this project was to pass the adversarial images through another CIFAR10 network to see if the attack could translate between architectures. The "simple deep CNN" model by the Keras developers was chosen due to its programming accessibility.

A random selection of images did not reveal many examples for which an adversarial image was misclassified by the simple deep CNN:

Original: truck (confidence=0.991)
Adversarial: truck (confidence=0.985)

Original: dog (confidence=0.272)
Adversarial: deer (confidence=0.284)

An evaluation of the original and adversarial test sets revealed that the adversarial images decreased accuracy by 2.57%.

Simple Deep CNN Test Set Evaluation		
	Original Set	***Adversarial Set***
Accuracy	0.7545	0.7288
Loss	0.7730	0.8538

Accuracy = correctly classified samples / total samples

For the subset of images where the adversarial counterpart failed to misclassify, the mean confidence of the original classification was 0.761. For the subset where the adversarial counterpart succeeded in misclassifying, the mean confidence of the original classification was 0.398. This showed that the adversarial example was more likely to succeed if the original classification had a low confidence level.

The adversarial examples influenced the accuracy of the simple deep CNN, but only by a small 2.57%. In a future project, it would be interesting to see what parameters of the FGS attack would affect the mean original confidence at which the adversarial counterparts successfully misclassify in the simple deep CNN.

V. Adversarial Training on Simple Deep CNN

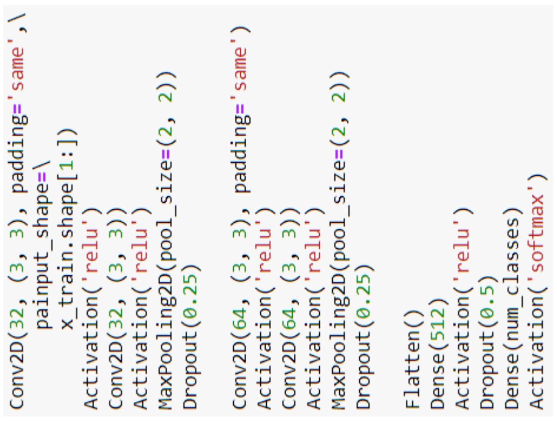
Having evaluated the adversarial test set on the simple deep CNN, it was time to retrain the network on the adversarial training set. This process is known as adversarial training, and it is used to defend against misclassification from adversarial images. I will refer to this newly trained network as the "retrained network."

Results showed that the retrained network had higher accuracy on both the original and adversarial test sets. Additionally, the decrease in accuracy from adversarial images was lower than with the original network, from a 2.57% decrease to a 1.11% decrease.

Retrained Simple Deep CNN Test Set Evaluation		
	Original Set	***Adversarial Set***
Accuracy	0.7685	0.7574
Loss	0.6887	0.7212

Accuracy = correctly classified samples / total samples

Original vs. Retrained Network		
	Original	***Retrained***
Accuracy loss from adversarial images	0.0257	0.0111



```
Conv2D(32, (3, 3), padding='same',\
    painput_shape=\
      x_train.shape[1:])
Activation('relu')
Conv2D(32, (3, 3))
Activation('relu')
MaxPooling2D(pool_size=(2, 2))
Dropout(0.25)

Conv2D(64, (3, 3), padding='same')
Activation('relu')
Conv2D(64, (3, 3))
Activation('relu')
MaxPooling2D(pool_size=(2, 2))
Dropout(0.25)

Flatten()
Dense(512)
Activation('relu')
Dropout(0.5)
Dense(num_classes)
Activation('softmax')
```

Fig. 2. Simple Deep CNN

This test showed that adversarial training was a viable defense against adversarial images. The cost of this defense was retraining the network, which on the EC2 p2.xlarge instance took 36 minutes. ML engineers must weigh the benefit of reduced accuracy loss against the cost of retraining a network. Additionally, they must consider that this is a reactive defense and not a proactive one, since it is impossible to know what kind of adversarial images a network will encounter before it encounters them.

VI. Gaussian Filtering

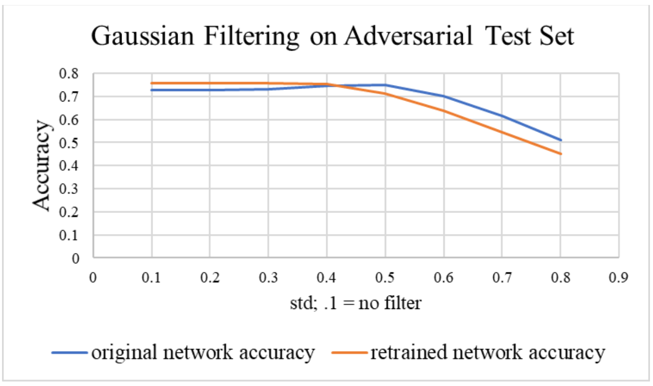
As a defense method against adversarial ML, adversarial training has a high computational cost. The last goal of this project was to evaluate the effectiveness of a lower cost defense method. I hypothesized that using some form of low pass filter on the adversarial images could reduce the effect of adversarial noise and would have a low computational cost. For that reason and ease of use, a gaussian filter was chosen. The SciPy gaussian filter used in this project is implemented as a sequence of one-dimensional convolution filters.

No filter
Predicted: truck (confidence=0.917)

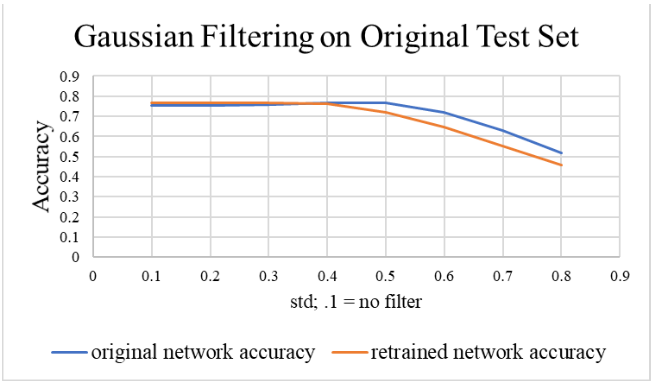
std=0.6
Predicted: truck (confidence=0.687)

std=0.8
Predicted: deer (confidence=0.911)

As seen above, the gaussian filter takes standard deviation for the gaussian kernel as an argument. The higher the standard deviation, the stronger the effect. To assess the effect of the filter on accuracy, accuracy tests were done with many standard deviation values on both test sets with both networks.



The situation in which we desire the gaussian filter to increase accuracy is on the adversarial test set with the original network. As seen in the results, from “no filter” to “std=0.5” on the adversarial test set with the original network, the accuracy goes from 0.7288 to 0.7492, an increase of 2.04%. After that, increasing the standard deviation of the filter decreases accuracy. This shows that gaussian filtering is a viable strategy to combat adversarial ML. Additionally, applying this filter takes under a thousandth of a second and can be done on the fly. It provides a basic and computationally cheap way to defend against adversarial images.



Two interesting trends to note in the data are that the filter never increases the accuracy of the retrained network, and the retrained network accuracy goes below the original network accuracy for both test sets after std=0.5. The first trend is interesting because it raises questions about why the filter could only decrease the accuracy. One potential cause is that adversarial training could overfit the model, making it less able to distinguish images that have gone through a low pass filter. The second trend ties into the first because if the first hypothesis were true, then the original network would perform better on blurry images. These are good questions to investigate in a future project.

VII. Conclusions

In this project I successfully implemented an untargeted fast gradient sign attack on a CIFAR10 EfficientNet, tested it on a simple deep CNN, used it to generate an adversarial image set, and then compared the defense effectiveness and computational cost of adversarial training and gaussian filtering. One result that I did not expect was for adversarial training to raise overall accuracy along with defensive capabilities against adversarial images. There might be a possible application for that in improving the accuracy of other networks. Another result that I did not expect was for the simple deep CNN architecture to defend so well against the untargeted FGS attack. I thought that because the attack was developed on a more complicated network than the simple deep CNN, that would somehow also make the attack misclassify any image for the simple deep CNN. In reality, the adversarial images only slightly reduced the accuracy on the different network. That was the least effective experiment in this project, but everything else worked well. I am curious how the defenses I used would work against a more effective attack on the simple deep CNN. For this project, the gaussian filter ended up being the easier and cheaper option.

VIII. Challenges

The challenges I faced in this project all stemmed from it being in a subject area I hadn't worked with before. My lack of knowledge led me to constant dead ends, bugs, and technical problems, but I'm glad I had the opportunity to do it because it has given me a taste of machine learning.

IX. References

[1] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," Dec. 2014.

[2] M. Ozdag, "Adversarial Attacks and Defenses Against Deep Neural Networks: A Survey," *Procedia Computer Science*, vol. 140, pp. 152–161, 2018.

[3] J. Su, D. V. Vargas, and K. Sakurai, "One Pixel Attack for Fooling Deep Neural Networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828–841, 2019.

[4] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," Dec. 2013.

[5] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," May 2019.