Adversarial Attacks on Neural Nets, and Distillation Defense for Mitigation

Rucha Sathe
Georgia Institute of Technology
Atlanta, GA
rsathe8@gatech.edu

Meghana Deepak
Georgia Institute of Technology
Atlanta, GA
mdeepak3@gatech.edu

Aditya Milind Pansare
Georgia Institute of Technology
Atlanta, GA
apansare6@gatech.edu

Abstract

Exploring deep learning as a novice often shapes our thought process in a way such that we think deep learning models while being rather BlackBox, are an extremely safe bet to make sure we get effective results. We make models complex and tune hyperparameters to make sure that we achieve higher accuracies, believing in the notion that our models are accurate, consistent, and cannot be tampered with. But as we explore further, we notice that our deep learning models can fall prey to a series of attacks that can disrupt the way our model works, and can sometimes even have a targeted & malicious effect. These are called adversarial attacks. and in this project, we want to explore how adversarial attacks impede a deep learning model's ability to classify images.

1. Introduction

The use of deep learning models have been ubiquitous nowadays, and help us solve business and research problems that deal with complex images, videos, audio files, and text files, among other. We invest a lot of resources to make sure that we have complex models that perform well. But these models are far from secure. Disrupting or impeding the way a model would work in cases of a critical task could cause a huge loss financially as well as stall tasks from being complete.

These disruptions could be brought about by conducting a series of ”attacks” which are essentially perturbations induced in the data samples or inputs. These attacks are called adversarial attacks. These are often considered the most common reason that can cause a malfunction in a model. It could be initiated as the model is training or can even be engineered to maliciously modify the input data to deceive a fully trained model.

These attacks can be be prevented by setting up defense mechanisms that can either be proactive or reactive in nature. Setting up an effective defence would help us protect these models and not let adversarial attacks affect critical business tasks. Hence, understanding different types of adversarial attacks and setting up defense mechanisms to prevent them are of paramount of importance in a deep learning deployment pipeline.

2. Motivation

Lately, the subject of adversarial attacks and defenses against them has attracted a lot of attention from the deep learning community. It is important to recognize the security threats posed by adversarial attacks and address ways in which they can be prevented or, at least, reduced. Several businesses and organizations, state or private, have been deploying state-of-the-art deep learning and machine learning models to aid and accomplish complex and critical tasks. Adversarial attacks can impede their model performance, or can even end up having effects of criminal nature. Our analysis of the defense methods will help us prevent these undesired effects.

In the recent past, a significant amount of research has been carried out on this topic. Some notable contributions like [3] presents an effective defense against adversarial attacks on deep learning classifier models. This two-pronged approach leverages both proactive as well as reactive methods of defense but is not immune to the transferring attack. [5] proposes a method which uses collaborative multi-task training to defend against a variety of adversarial attacks. While all these publications demonstrate great progress in the field, [4] stands out in terms of the concept that they have implemented. The method proposed in this paper is based on the process of distillation. In this approach, knowledge extracted from a deep neural network is used to strengthen itself against adversarial samples. We have tried to explore this concept further by experimenting on different datasets, analyzing and comparing the results of the attacks and defenses on each of them.

3. Datasets

For the purpose of the project, we have trained our attack-and-defense model on two different datasets, namely, FashionMNIST and KMNIST.

Fashion-MNIST [6] is a dataset of Zalando's article images, consisting of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes. The classes represented by this dataset are as follows: T-shirt/top (0), Trouser (1), Pullover (2), Dress (3), Coat (4), Sandal (5), Shirt (6), Sneaker (7), Bag (8), Ankle boot (9). The motivation behind creating this dataset was the fact that MNIST is comparatively very easy to train on and make predictions. Also, since MNIST is widely used, FashionMNIST aims at providing diversity to help train deep learning networks on new and more complicated data.

The second dataset we have used is KMNIST [1]. KMNIST (Kuzushiji-MNIST or Cursive hiragana-MNIST) was introduced as an alternative to MNIST. It contains images with the first entries from the 10 main Japanese hiragana character groups. Similar to FashionMNIST, KMNIST consists of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes. [2]

4. Approach

Implemented a network to train the data in which three non-targeted adversarial attacks and a defense mechanism to counterattack was performed. The three attacks implemented primarily target image perturbations as we intend to retain the model weights as a constant and the following section elaborates the attacks further.

Fast Gradient Sign Method

The addition of noise in the direction of the gradient of the loss function on the image data results in an almost identical image (adversarial image) which increases the loss further increasing model error and in turn leads to an incorrect prediction. An interesting feature of the method is computation of gradients with respect to the input image which aids in the creation of an image that maximizes the loss. One popular method of implementation is evaluating the contribution of each pixel to the loss value and then with suitable perturbation the attack is generated. The magnitude of noise induced is scaled using the epsilon parameter constrained by a max norm. The below expression summarizes the attack where

$$x^{adv} = x + \varepsilon \cdot \operatorname{sign}\left(\nabla_x J\left(x, y_{\text{true}}\right)\right))$$

where

x – Input Image

x^{adv} – Adversarial Image

J – Loss Function

y_{true} – Model Output for x

ε – Tunable Parameter

Iterative Fast Gradient Sign Method(I-FGSM)

An iterative implementation of the FGSM with specific step sizes and trimming pixel values of interim results post each step in order to ensure the are in the likely neighborhood of the original image is performed as a part of this attack.

$$x_0^{adv} = x, \quad x_{N+1}^{adv} = \operatorname{Clip}_{x,\varepsilon}\left\{x_N^{adv} - \alpha \cdot \operatorname{sign}\left(\nabla_x J\left(x_N^{adv}, y_{LL}\right)\right)\right\}$$

where

x – Input Image

x_i^{adv} – Adversarial image at i

J – Loss Function

step

y_{LL} – Least Likely Class

ε – Tunable Parameter

α – Step Size

The number of iterations are selected based on heuristics and considering sufficiency of the attack to attain an edge of max normal but constrained enough to reduce the computational cost of the iterations. A thorough analysis states that the iterative methods are more powerful white-box adversaries over single step adversaries but at the risk of worse transfer-ability. Additionally, this method works sufficiently for MNSIT and CIFAR10 datasets as the classes are highly distinct and lesser in number.

Momentum Iterative Fast Gradient Sign Method(MI-FGSM)

While in I-FGSM the iterative calculation of gradients happens at each step and then the gradient result is added to the attacked sample, this results in only being able to attack white-box models directly and not black box models which restricts a majority of it's practical applications. In several cases, they can greedily overfit specific network parameters. This method uses the integration of momentum as a term in the adversarial attack process which ensures stability while updating directions to avoid poor local maxima. This results in better attacks and removal of extreme local values and shock updates. The process of updating is close to I-FGSM and results in the following equation

$$g_{t+1} = \mu g_t + \frac{\nabla_x J(x_t^*, y)}{\|\nabla_x J(x_t^*, y)\|_1}$$
$$x_t^* = x_{t-1} + \alpha * \text{sign}(g_{t+1})$$

where μ is the decay factor of the momentum term and gn is the accumulated gradient at iteration n.

Distillation

To counterattack the adversarial attack introduced a defense called the defensive distillation which helps reduce the attack on the image samples. The distillation process results in reduction of gradients used in adversarial samples by a huge factor and increases features required for attacks to be more effective on the samples. Thus making it a robust and easily generalizable method to defense such attacks.

Steps in Distillation

Implemented the distillation defense techniques as follows Trained the network F on the given training set (X,Y) on setting the temperature of the softmax to T. Calcuated the scores (post softmax operation) given by F(X) repeatedly and computed the values at temperature T. Trained another network F'T with softmax at temperature T on the same dataset with soft labels (X,F(X)). Referring to the model FT as the distilled model and Using the distilled network F'T along with softmax at temperature 1,represented as F'1 while making predcitions on test data Xtest(or adversarial examples).

5. Experimentation Results

Before we detail our experiments, we would like to define the class labels and what they point to.

For KMNIST Dataset, we use 10 class numbers (0 to 9) which denote a Japanese character as the class label. For FashionMNIST Dataset, we use the same class numbers to denote an apparel or a garment as the class label. Below is a table that shows the mapping for each class number to corresponding label in KMNIST and FashionMNIST.

Taking inspiration from the work described in [4], we used two separate networks for the attack-and-defense process. Both networks use two convolutional layers and two linear layers, interspersed with ReLU and Maxpool layers according to a typical deep learning network. The input is a batch of grayscale images (1-channel) of dimensions 28 * 28 pixels. After being passed through the neural network, the last linear layer represents a classifier, that is, its output size is equivalent to the number of classes of the dataset. For the purpose of the attack, we first introduce perturbations to images in accordance with the three methods mentioned previously, namely, FGSM, IFGSM and MIFGSM. Next, we pass these modified images through the trained network using a large range of epsilon values: **0, 0.007, 0.01, 0.02, 0.03, 0.05, 0.1, 0.2 and 0.3**. Next, we run the distillation defense on all three attacks one by one.

Class Number	KMNIST	FashionMNIST
0	o	T-shirt/Top
1	ki	Trouser
2	su	Pullover
3	tsu	Dress
4	na	Coat
5	ha	Sandal
6	ma	Shirt
7	ya	Sneaker
8	re	Bag
9	wo	Ankle Boot

Table 1. KMNIST and FashionMNIST labels corresponding to the 10 class numbers

Below are a couple of figures that show the accuracy v/s ϵ plot and some examples misclassified images for FGSM.

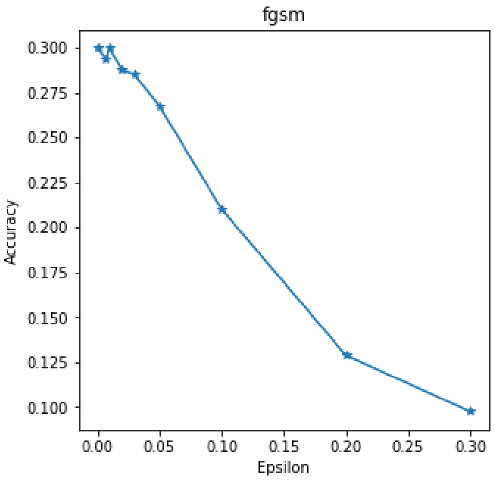

Figure 1. Accuracy vs ϵ plot for FGSM attack on KMNIST

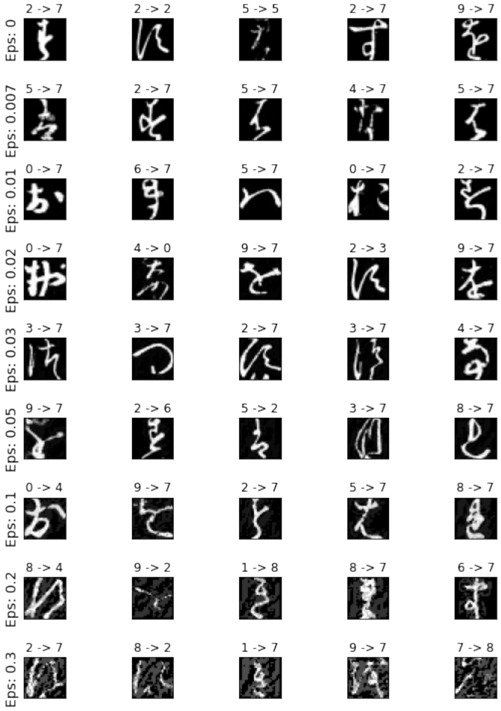

Figure 2. Misclassified examples for FGSM attack on KMNIST

Now, we implement the distillation defense as shown in the Approach section. Below are the improved results.

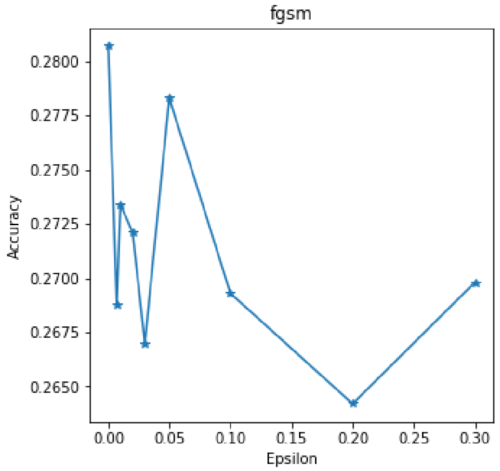

Figure 3. Accuracy vs ϵ plot for distillation defense against FGSM attack on KMNIST

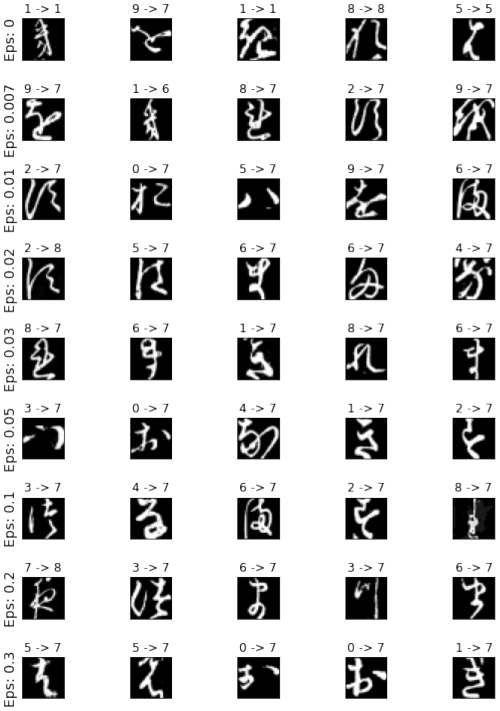

Figure 4. Misclassified examples for distillation defense against FGSM attack on KMNIST

Now, we show the same series of results for FashionMNIST.

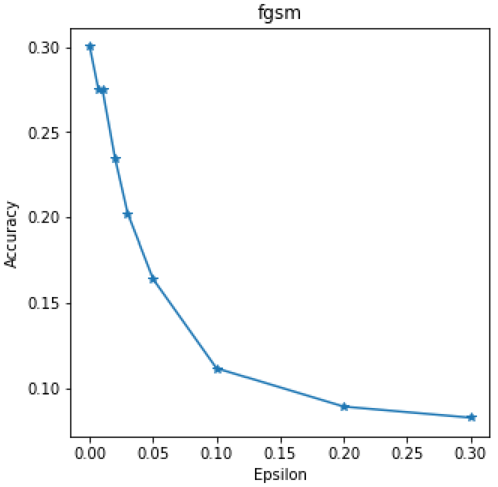

Figure 5. Accuracy vs ϵ plot for FGSM attack on FashionMNIST

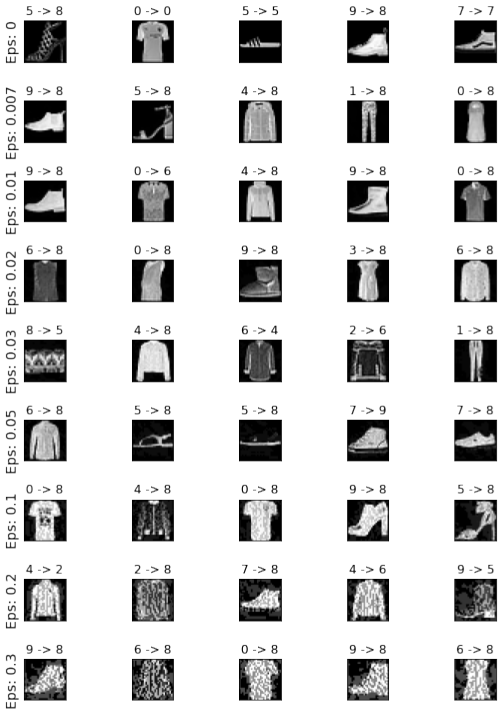

Figure 6. Misclassified examples for FGSM attack on FashionMNIST

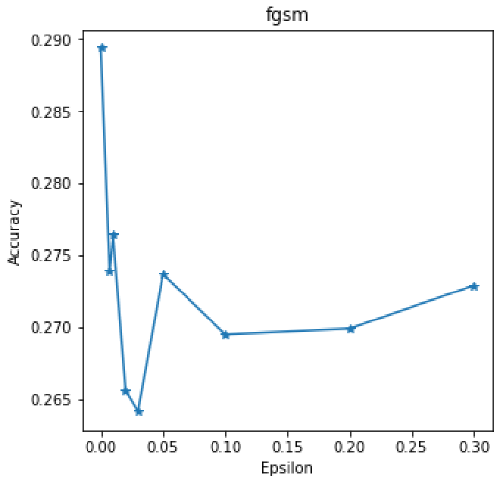

Figure 7. Accuracy vs ϵ plot for distillation defense against FGSM attack on FashionMNIST

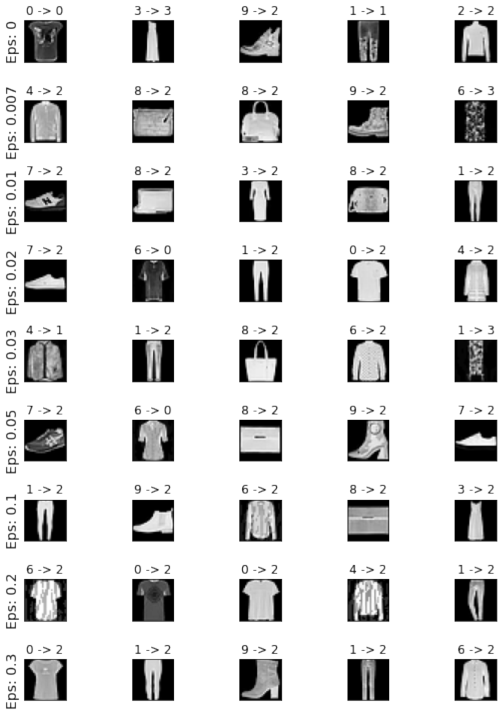

Figure 8. Misclassified examples for distillation defense against FGSM attack on FashionMNIST

6. Conclusion

We experimented on two datasets KMNIST and FashionMNIST using the same three methods of attack, namely FGSM, I-FGSM and MI-FGSM. Following this, we carried out distillation defense on both the datasets affected by the three above mentioned attacks. We observed that distillation defense is effective against all three types of attacks and achieved a significant amount of success for all the attacks. In conclusion, we deduced that distillation defense is able to protect the network best against FGSM as compared to I-FGSM and MI-FGSM.

7. Acknowledgments

We are grateful to Prof. Zsolt Kira for giving us the opportunity to closely study and work on the topic of adversarial attacks and distillation defenses. We are also grateful to the TAs for their constant support and guidance throughout the project.

8. Team Contributions

	Member	Contribution
1	Rucha Sathe	Researched & Implemented FGSM attack. Co-implemented Distillation Defense. Researched and onboarded datasets KMNIST & FashionMNIST.
2	Meghana Deepak	Implemented the base model. Researched & Implemented I-FGSM attack. Co-implemented Distillation Defense. Analyzed the results obtained.
3	Aditya Milind Pansare	Researched & Implemented MI-FGSM attack. Co-implemented Distillation Defense. Conducted experimentation and testing for the different datasets.

References

[1] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature, 2018.

[2] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna M. Wallach, Hal Daumé III, and Kate Crawford. Datasheets for datasets. *CoRR*, abs/1803.09010, 2018.

[3] Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. *CoRR*, abs/1705.09064, 2017.

[4] Nicolas Papernot, Patrick D. McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. *CoRR*, abs/1511.04508, 2015.

[5] Derek Wang, Chaoran Li, Sheng Wen, Yang Xiang, Wanlei Zhou, and Surya Nepal. Defensive collaborative multi-task training - defending against adversarial attack towards deep neural networks. *CoRR*, abs/1803.05123, 2018.

[6] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.