Narrow attack comprehensive defense

Brian Rikshpun

Guy Cohen

Ester Vaknin

Almog Klein

Afeka College of Engineering

February 11, 2021

1 Abstract

Over the years, the field of computer vision became the workhorse of contemporary AI technologies with a wide variety of applications ranging from self-driving cars to surveillance and security. While Convolution neural networks (CNN) achieve high preferences on image classification tasks, recent studies show that they are vulnerable to adversarial attacks in the form of subtle perturbations, often too small to be perceptible, that lead a model to predict incorrect outputs [1].

The defenses against the adversarial attacks are being developed along with three main directions: using modified training or modified input, modifying networks, and using external models as network add-ons. In this work, we will focus on the first direction, to be specific, brute-force adversarial training.

In this work, we tried to find out how different attacked training data combining different percent of adversarial data will affect the robustness of the CNN model accuracy against different adversarial attacks. We were surprised to find out that the results of our research points that training with a specific attack will make the model robust to different attacks which he never saw.

2 Introduction

Several machine learning models, including state-of-the-art neural networks, are vulnerable to adversarial examples. That is, these machine learning models misclassify examples that are only slightly different from correctly classified examples drawn from the data distribution.

While there are ‘easy to make’ attacks such as FGSM which requires us to forward propagate the image, then backpropagate its gradient sign (+/-) concerning the model architecture and weights after initial training and give the whole pixels in the picture a very small nudge so there is no visible difference after the attack. since we are moving the points concerning the gradient direction, we are attempting to maximize the loss function and by that, increase the probability to misclassify the picture [2].

There are also more complex attacks, such as one-pixel attack [3] which requires us to find a solution to an optimization problem using Differential Evolution algorithm that will tell us eventually the one pixel we need to change, and the color of the pixel after the change which will give us the best perturbation.

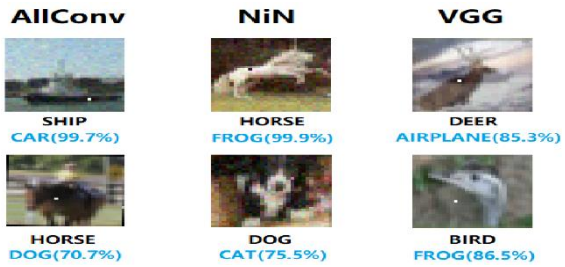

Figure 1 - Example of one pixel attack

Recent studies have shown that attackers can easily generate a malicious sample by adding a small perturbation to a normal sample. To overcome this problem, various mechanisms have been developed to defense against adversarial attacks on deep neural networks. It is customary to divide the developed mechanisms into three main types [1].

The first type focus on using modified training during learning or modified input during testing. The most prominent of the mechanisms associated with this type called 'Brute-Force adversarial training'. This is a brute force solution where we simply generate a lot of adversarial examples and explicitly train the model not to be fooled by each of them. Adversarial training is offered as the first line of defense against attacks and improves robustness of the networks [4]. In our research we will examine the behavior and the quality of this defense mechanism.

Unlike the first type of defense mechanisms we have reviewed, the other two types are more concerned with the neural networks themselves [1]. There is a type of defense mechanisms that is characterized as ‘modifying’ a network, by adding additional layers or sub-networks, changing loss or activation functions, etc. These mechanisms make changes to the original deep neural network architecture or parameters during training. In this type, we will describe a new defensive approach for DNN models called 'DeepCloak'. The method based on inserting a mask layer right

before the linear layer handling classification. The mask layer serves as a selector, which will keep the necessary features and remove the unnecessary features [5]. The method has been proven to be a computationally efficient and by removing a small percent of features the adversarial robustness can be greatly improved, and the model still achieves high accuracy.

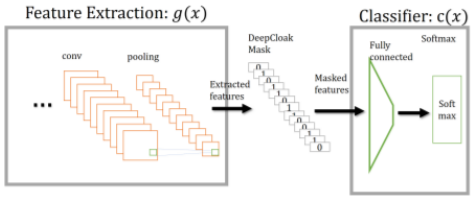

Figure 2 - A sketch of DeepCloak

Moreover, the last type of defense mechanism defined as 'network add-on', using external models when classifying unseen examples. Unlike the previous type, this keeps the original model intact and appends external models to it during testing. We will focus on one prominent mechanism of this type which is defined as 'defense against universal perturbations'. This mechanism is based on a Perturbation Rectifying Network (PRN) as 'pre-input' layers to a targeted model. The PRN is learned from real and synthetic image-agnostic perturbations and a perturbation detector is separately trained [6]. The proposed framework rectifies the images to restore the network predictions. This mechanism has been found to be very effective, it acts as an external wrapper such that the PRN and the detector trained to counter the adversarial attacks can be kept secretive in order refrain from potential counter-counter attacks.

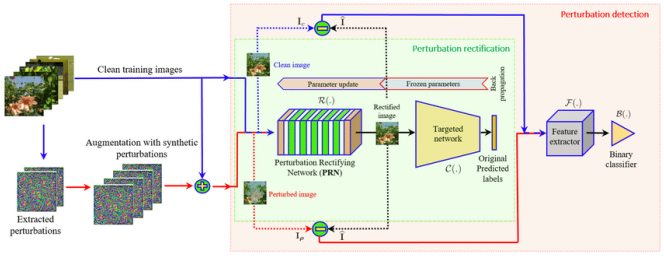

Figure 3 - Training schematics of PRN

3 Data description

In our work we used the MNIST image datasets, the MNIST dataset includes examples of handwritten digits (0 – 9), 60,000 training examples, and 10,000 test examples.

Figure 4 - MNIST dataset

4 Methodology of work

Our work is based on a code we found online – Link
Our goal in this work is to find out how does training a CNN classification model with attacked data affects the robustness of the CNN against other attacks. The attacks we used in our research are FFGSM, FGSM, PGD, UPGD, AutoAttack, and DeepFool. For each attack, we trained the same CNN model but on a different percentage of attacked data: 20k adversarial examples (33.3%), 30k adversarial examples (50%), and 60k adversarial examples (100%) each iteration we used the same attack and tested the model with the same attack and the other five.

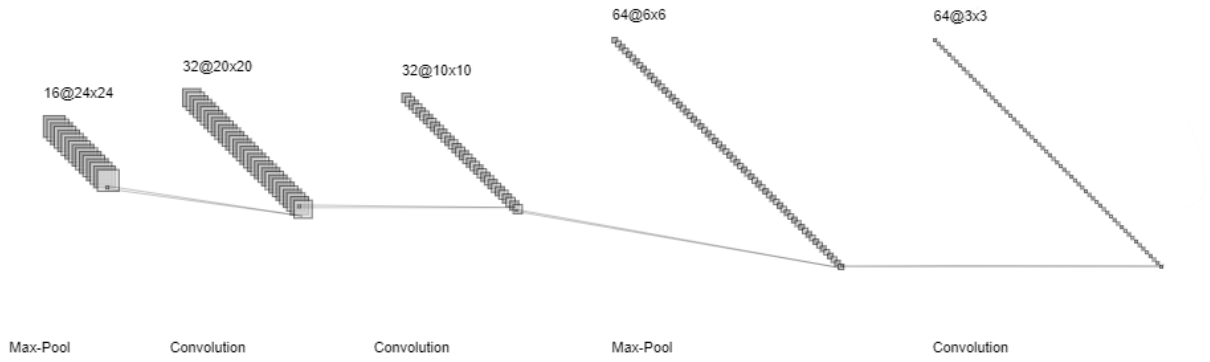

Figure 5 - Our CNN architecture

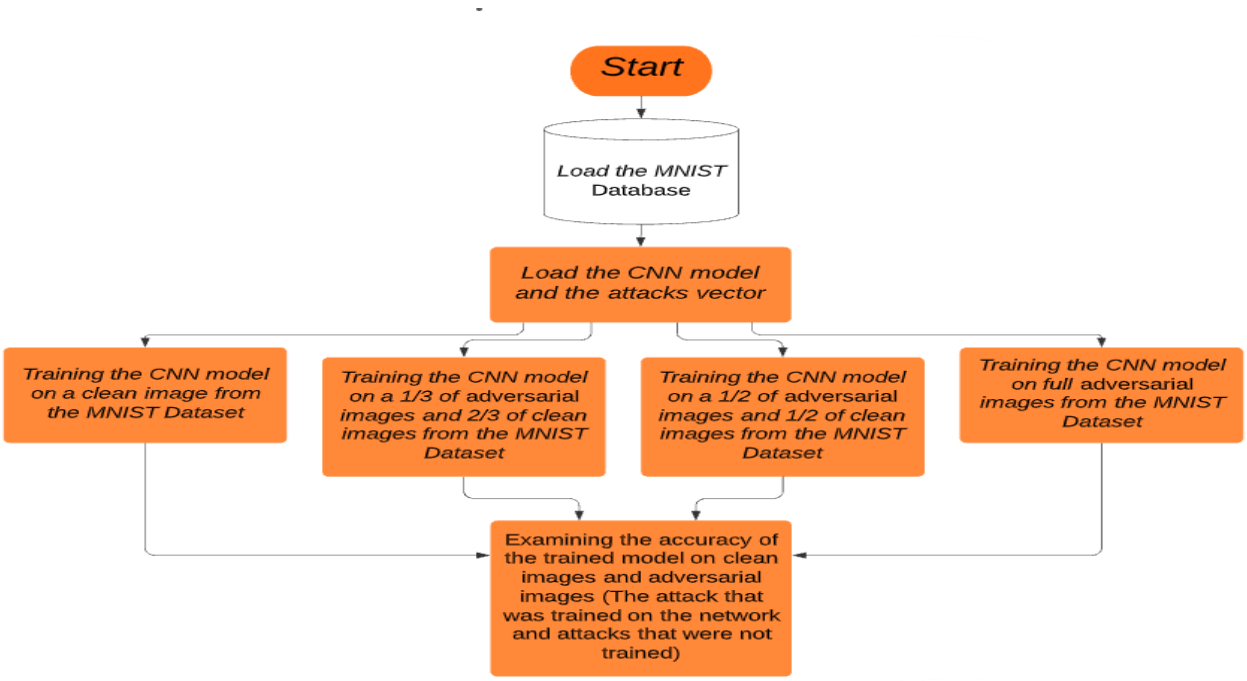

Figure 6 - Work methodology

5 Results

The first step we took was to train and test the CNN on clean image data, the result was 97.35% accuracy for 1 epoch, 98.98% accuracy for 10 epochs and 99.44% accuracy for 100 epochs. Then, we tested the same CNN accuracy on new attacked data which will be used as the benchmark:

Attack	CNN model accuracy (1 epoch)	CNN model accuracy (10 epochs)	CNN model accuracy (100 epochs)
FFGSM	81.65%	85.73%	92.6%
FGSM	78.73%	83.05%	91.62%
PGD	75.65%	69.14%	65.94%
UPGD	74.82%	-	-
AutoAttack	73.82%	-	-
DeepFool	41.81%	-	-

Table 1 - Clean test, attacked train results

After this we executed our work methodology as mentioned above, the results (link to the dashboard):

full advatk train attack: FFGSM							50/50 train attack: FFGSM							33/67 train attack: FFGSM						
98.31						NN clean accuracy full advatk train:	98.62						CNN clean accuracy train5050:	98.41						CNN clean accuracy 33/67 train
DeepFool	AutoAttack	UPGD	PGD	FGSM	FFGSM	NEW ATTACKS TEST	DeepFool	AutoAttack	UPGD	PGD	FGSM	FFGSM	NEW ATTACKS TEST	DeepFool	AutoAttack	UPGD	PGD	FGSM	FFGSM	NEW ATTACKS TEST
95.85	97.35	96.79	96.8	96.99	**97.29**	attacked accuracy	94.33	96.8	95.95	95.95	96.1	**96.64**	attacked accuracy	97.94	97.89	97.84	97.86	97.87	**97.87**	attacked accuracy
full advatk train attack: FGSM							**50/50 train attack: FGSM**							**33/67 train attack: FGSM**						
97.27						NN clean accuracy full advatk train:	98.07						CNN clean accuracy train5050:	98.6						CNN clean accuracy 33/67 train
DeepFool	AutoAttack	UPGD	PGD	**FGSM**	FFGSM	NEW ATTACKS TEST	DeepFool	AutoAttack	UPGD	PGD	**FGSM**	FFGSM	NEW ATTACKS TEST	DeepFool	AutoAttack	UPGD	PGD	**FGSM**	FFGSM	NEW ATTACKS TEST
95.84	97.59	97.73	97.7	**97.95**	98.2	attacked accuracy	94.36	96.31	95.46	95.5	**95.6**	96.2	attacked accuracy	98.11	98.09	98.07	98.11	**98.09**	98.13	attacked accuracy
full advatk train attack: PGD							**50/50 train attack: PGD**							**33/67 train attack: PGD**						
98.26						NN clean accuracy full advatk train:	98.59						CNN clean accuracy train5050:	98.2						CNN clean accuracy 33/67 train
DeepFool	AutoAttack	UPGD	**PGD**	FGSM	FFGSM	NEW ATTACKS TEST	DeepFool	AutoAttack	UPGD	**PGD**	FGSM	FFGSM	NEW ATTACKS TEST	DeepFool	AutoAttack	UPGD	**PGD**	FGSM	FFGSM	NEW ATTACKS TEST
95.84	97.68	97.4	**97.3**	97.42	97.67	attacked accuracy	95.57	97.18	96.45	**96.41**	96.4	97.01	attacked accuracy	98.09	98.07	98.04	**97.98**	98.05	98.02	attacked accuracy
full advatk train attack: UPGD							**50/50 train attack: UPGD**							**33/67 train attack: UPGD**						
96.65						NN clean accuracy full advatk train:	98.42						CNN clean accuracy train5050:	97.3						CNN clean accuracy 33/67 train
DeepFool	AutoAttack	**UPGD**	PGD	FGSM	FFGSM	NEW ATTACKS TEST	DeepFool	AutoAttack	**UPGD**	PGD	FGSM	FFGSM	NEW ATTACKS TEST	DeepFool	AutoAttack	**UPGD**	PGD	FGSM	FFGSM	NEW ATTACKS TEST
95.44	97.84	**98.67**	98.5	98.77	98.91	attacked accuracy	95.63	96.96	**96.3**	96.27	96.28	96.81	attacked accuracy	97.88	97.87	**97.79**	97.77	97.77	97.82	attacked accuracy
full advatk train attack: AutoAttack							**50/50 train attack: AutoAttack**							**33/67 train attack: AutoAttack**						
98.07						NN clean accuracy full advatk train:	98.04						CNN clean accuracy train5050:	98.62						CNN clean accuracy 33/67 train
DeepFool	**AutoAttack**	UPGD	PGD	FGSM	FFGSM	NEW ATTACKS TEST	DeepFool	**AutoAttack**	UPGD	PGD	FGSM	FFGSM	NEW ATTACKS TEST	DeepFool	**AutoAttack**	UPGD	PGD	FGSM	FFGSM	NEW ATTACKS TEST
92.91	**96.61**	95.97	96	95.99	96.52	attacked accuracy	89.77	**95.33**	94.06	94.14	94.2	94.94	attacked accuracy	97.48	**97.49**	97.18	97.27	97.21	97.3	attacked accuracy
full advatk train attack: DeepFool							**50/50 train attack: DeepFool**							**33/67 train attack: DeepFool**						
97.36						NN clean accuracy full advatk train:	97.81						CNN clean accuracy train5050:	98.65						CNN clean accuracy 33/67 train
DeepFool	AutoAttack	UPGD	PGD	FGSM	FFGSM	NEW ATTACKS TEST	**DeepFool**	AutoAttack	UPGD	PGD	FGSM	FFGSM	NEW ATTACKS TEST	**DeepFool**	AutoAttack	UPGD	PGD	FGSM	FFGSM	NEW ATTACKS TEST
94.82	95.53	94.92	94.9	94.96	95.34	attacked accuracy	**95.49**	95.04	93.97	94.11	94.13	94.92	attacked accuracy	**97.73**	97.74	97.68	97.78	97.72	97.68	attacked accuracy

Table 2 – Training with attacked data results 1 epoch

full advatk train attack: FFGSM				50/50 train attack: FFGSM				33/67 train attack: FFGSM			
97.38			CNN clean accuracy full advatk train:	98.75			CNN clean accuracy 50/50 train:				CNN clean accuracy 33/67 train:
PGD	FGSM	**FFGSM**	NEW ATTACKS TEST	PGD	FGSM	**FFGSM**	NEW ATTACKS TEST	PGD	FGSM	**FFGSM**	NEW ATTACKS TEST
99.15	99.36	**99.42**	attacked accuracy	98.09	98.23	**98.32**	attacked accuracy	97.18	97.36	**97.87**	attacked accuracy
full advatk train attack: FGSM				**50/50 train attack: FGSM**				**33/67 train attack: FGSM**			
97.91			CNN clean accuracy full advatk train:	98.95			CNN clean accuracy 50/50 train:				CNN clean accuracy 33/67 train:
PGD	**FGSM**	FFGSM	NEW ATTACKS TEST	PGD	**FGSM**	FFGSM	NEW ATTACKS TEST	PGD	**FGSM**	FFGSM	NEW ATTACKS TEST
99.02	**99.3**	99.29	attacked accuracy	98.58	**98.81**	98.95	attacked accuracy	98.13	**98.24**	98.38	attacked accuracy
full advatk train attack: PGD				**50/50 train attack: PGD**				**33/67 train attack: PGD**			
96.09			CNN clean accuracy full advatk train:	98.74			CNN clean accuracy 50/50 train:				CNN clean accuracy 33/67 train:
PGD	FGSM	FFGSM	NEW ATTACKS TEST	**PGD**	FGSM	FFGSM	NEW ATTACKS TEST	**PGD**	FGSM	FFGSM	NEW ATTACKS TEST
99.48	99.35	99.48	attacked accuracy	**98.5**	98.58	98.72	attacked accuracy	**98.35**	98.34	98.6	attacked accuracy

Table 3 – Training with attacked data results 10 epoch

full advatk train attack: FFGSM				50/50 train attack: FFGSM				33/67 train attack: FFGSM			
99.02			CNN clean accuracy full advatk train:	99.01			CNN clean accuracy 50/50 train:				CNN clean accuracy 33/67 train:
PGD	FGSM	**FFGSM**	NEW ATTACKS TEST	PGD	FGSM	**FFGSM**	NEW ATTACKS TEST	PGD	FGSM	**FFGSM**	NEW ATTACKS TEST
99.26	99.32	**99.38**	attacked accuracy	98.47	98.66	**98.68**	attacked accuracy	98.71	98.74	**98.96**	attacked accuracy
full advatk train attack: FGSM				**50/50 train attack: FGSM**				**33/67 train attack: FGSM**			
99.01			CNN clean accuracy full advatk train:	99.31			CNN clean accuracy 50/50 train:				CNN clean accuracy 33/67 train:
PGD	**FGSM**	FFGSM	NEW ATTACKS TEST	PGD	**FGSM**	FFGSM	NEW ATTACKS TEST	PGD	**FGSM**	FFGSM	NEW ATTACKS TEST
99.32	**99.42**	99.42	attacked accuracy	98.9	**99.11**	99.11	attacked accuracy	98.62	**98.74**	98.93	attacked accuracy
full advatk train attack: PGD				**50/50 train attack: PGD**				**33/67 train attack: PGD**			
			CNN clean accuracy full advatk train:	98.99			CNN clean accuracy 50/50 train:				CNN clean accuracy 33/67 train:
PGD	FGSM	FFGSM	NEW ATTACKS TEST	**PGD**	FGSM	FFGSM	NEW ATTACKS TEST	**PGD**	FGSM	FFGSM	NEW ATTACKS TEST
			attacked accuracy	**99.18**	98.86	99.09	attacked accuracy	**98.66**	98.49	98.68	attacked accuracy

Table 4 – Training with attacked data results 100 epoch

6 Conclusions

Neural networks can be easily exposed to adversarial attacks. A very small change in the image (as we saw in the FSGM method) or even a single change in one of the pixels in the image (May not be recognizable by the human eye) can mislead the classification with very high model confidence.

We found out that brute force adversarial training is a very comfortable way to deal with those exposes. As the results show, small trained attacked data will improve the robustness of the model and will not affect the accuracy of the CNN model drastically as expected.

Moreover, we can see that for each attack we used to train the model, the robustness we got can also deal with other attacks (attacks the model didn't train on - never seen before), which makes us believe that the gained defense on the model is peripheral. We also noticed that training attacked data with at least 10 epochs will give us the same accuracy as the clean accuracy of the model.

Our research drawbacks are the variety of the dataset and optimization of the hyper parameters of each attack. We used MNIST dataset which is not as complex as the CIFAR or ImageNet, and trained the attacks with the same hyper parameters each time.

By looking at the results of our research, we recommend combining adversarial examples in the training process of the CNN, it can cost us a slight loss in the accuracy but a very effective robustness gain to the model we train. We also recommend making the adversarial in the easiest and fastest way since as we saw, there is no significant difference between the robustness gained from training the model on a specific attack against other attacks.

7 Reference

[1] Akhtar, Naveed, and Ajmal Mian. "Threat of adversarial attacks on deep learning in computer vision: A survey." *Ieee Access* 6 (2018): 14410-14430.

[2] Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." *arXiv preprint arXiv:1412.6572* (2014). (FGSM

[3] Su, Jiawei, Danilo Vasconcellos Vargas, and Kouichi Sakurai. "One pixel attack for fooling deep neural networks." *IEEE Transactions on Evolutionary Computation* 23.5 (2019): 828-841.

[4] C. Szegedy et al. (2014). ‘‘Intriguing properties of neural networks.’’ [Online]. Available: https://arxiv.org/abs/1312.6199

[5] J. Gao, B. Wang, Z. Lin, W. Xu, and Y. Qi. (2017). ‘‘DeepCloak: Masking deep neural network models for robustness against adversarial samples.’’ [Online]. Available: https://arxiv.org/abs/1702.06763

[6] N. Akhtar, J. Liu, and A. Mian. (2017). ‘‘Defense against universal adversarial perturbations.’’ [Online]. Available: https://arxiv.org/abs/1711.05929