ARE DEEP NEURAL NETWORKS COMPLETELY ROBUST USING JACOBIAN REGULARIZATION?

CS5339 PROJECT

Lim Jia Xian, Clarence
A0212209U
e0503488@u.nus.edu

April 18, 2020

1 Introduction

It is known that Deep Neural Networks (DNN) are able to produce amazing results in various applications of computer vision. In fact, there are applications that have been deployed in real life scenarios that utilize Deep Convolution Neural Networks (CNN) for object detection and recognition tasks. Even though these networks are able to perform those tasks with high accuracy, it has been shown that they are highly vulnerable to adversarial attacks [1]. These attacks exploit the inherent fault in the network by making a small change in the input that would cause the network to deviate their prediction from the ground truth. In addition, the changes are usually negligible to human perception. DeepFool[2], utilize a simple algorithm to successfully trick many state-of-the-art DNN. By exploiting the classification decision boundaries, DeepFool is able to calculate the minimum perturbation needed to change the predicted class of the input. Due to this reason, providing robustness to adversarial attacks is an important challenge in networks training and has led to a wide range of research on this area. In this review, we seek to understand the following paper [3] which claims that using Jacobian regularization adds robustness to the model. We are also interested in answering the following question: *Are Deep Neural Networks completely robust by using Jacobian regularization?* In the following sections, we will describe the problem formally and provide one of the main contribution of the paper, then we shall conduct some experiments to verify and visualize some examples. Finally, we will provide our interpretation of the results and discuss the conclusion with future works.

2 Problem: Adversarial attacks

As most adversarial attacks on images are mostly negligible to human perception. We shall understand how they are done in this section. First, we shall define the following notation that is used by [3] to describe the problem in a formal way. They denote the input of the network as:

$$x_i \in \mathbb{R}^D, \qquad i = 1 \dots N, \qquad X = \begin{bmatrix} x_1^T \\ \vdots \\ x_N^T \end{bmatrix} \in \mathbb{R}^{N \times D}, \tag{1}$$

where the input x_i is a D-dimensional vector and the training dataset X consists of N training examples. The output of the network denoted as $f(x_i)$ is a K-dimensional vector, where $f(x_i) \in \mathbb{R}^K$. The output is obtained using a softmax operation as shown

$$f(x_i) = softmax\{z^{(L)}(x_i)\} \tag{2}$$

where $z^{(L)}(x_i)$ is the output of the last fully connected layer in the network for the input x_i. The index $l = 1, ..., L$ is used to specify a certain layer in a network with L layers and the output of the l^{th} layer of the network is represented as $z^{(l)}$ and the output of the k^{th} neuron in this layer is $z_k^{(l)}$.

In [2], the paper shown that adversarial attacks can be performed by first locally treating the decision boundaries as hyper-surfaces in the $K-$ dimensional output space of the network. They proposed the following lemma to approximates the distance between an input and a perturbed input classified to be at the boundary of a hyper-surface separating between the 2 probable classes of x, which is k_1 and k_2.

Lemma 1 *The first order approximation for the distance between an input x, with class k_1, and a perturbed input classified to the boundary hyper-surface separating the classes k_1 and k_2 for an ℓ_2 distance metric is given by*

$$d = \frac{|z_{k_1}^{(L)}(x) - z_{k_2}^{(L)}(x)|}{||\nabla_x z_{k_1}^{(L)}(x) - \nabla_x z_{k_2}^{(L)}(x)||_2} \tag{3}$$

Based on this lemma, [2] proposed the following corollary which provides a proxy for the minimal distance that may lead to fooling the network.

Corollary 2 *Let k^* be the correct class for the input sample x. Then the ℓ_2 norm of the minimal perturbation necessary to fool the classification function is approximated by*

$$d^* = \min_{k \neq k^*} \frac{|z_{k^*}^{(L)}(x) - z_k^{(L)}(x)|}{||\nabla_x z_{k^*}^{(L)}(x) - \nabla_x z_k^{(L)}(x)||_2} \tag{4}$$

With the d^* calculated, any trained network model with only ℓ_2 regularization applied can be easily fooled when d^* is applied to the input x as shown in [2].

3 Paper Contribution

One of the main contribution by [3] lies in providing a theoretically inspired novel approach to improve the robustness of the network. Their method applies the Frobenius norm to the Jacobian of the network as a regularization. The following notations were used by [3] to defined the Jacobian regularization.
The Jacobian matrix of the layer L evaluated at the point x_i is $J^{(L)}(x_i) = \nabla_x z_k^{(L)}(x_i)$ and similarly, $\nabla_x z_k^{(L)}(x_i)$is the k^{th} row in the Jacobian matrix $J^{(L)}(x_i)$. The network's Jacobian matrix is given by

$$J(x_i) \triangleq J^{(L)}(x_i) = \begin{bmatrix} \frac{\partial z_1^{(L)}(x_i)}{\partial x_1} & \cdots & \frac{\partial z_1^{(L)}(x_i)}{\partial x_D} \\ \vdots & \ddots & \vdots \\ \frac{\partial z_K^{(L)}(x_i)}{\partial x_1} & \cdots & \frac{\partial z_K^{(L)}(x_i)}{\partial x_D} \end{bmatrix} \in \mathbb{R}^{K \times D}, \tag{5}$$

where $x = (x_{(i)} \dots x_{(D)})^T$. Hence, the Jacobian regularization term which utilize the Frobenius norm for an input sample x_i is defined as

$$||J(x_i)||_F^2 = \sum_{d=1}^{D} \sum_{k=1}^{K} \left(\frac{\partial}{\partial x_d} z_k^{(L)}(x_i) \right)^2 = \sum_{k=1}^{K} ||\nabla_x z_k^{(L)}(x_i)||_2^2. \tag{6}$$

Using the regularization term in (6) with a standard cross-entropy loss function on the training data, the following loss function for training is defined as:

$$Loss = -\sum_{i=1}^{N} \sum_{k=1}^{K} y_{ik} \ log f_k(x_i) + \lambda \sqrt{\sum_{d=1}^{D} \sum_{k=1}^{K} \sum_{i=1}^{N} \left(\frac{\partial}{\partial x_d} z_k^{(L)}(x_i) \right)^2}, \tag{7}$$

where, λ is used to denote the hyper-parameter which controls the weight of the regularization penalty in the loss function and $y_i \in \mathbb{R}^K$ is a one-hot vector representing the correct class of the input x_i.

To justified the robustness of the Jacobian matrix, they first described the relationship between the adversarial perturbations and the Jacobian matrix of a network using the ℓ_2 distance metric as so:

$$\frac{||x_{pert} - x||_2}{||x_{same} - x||_2} \approx 1 \qquad and \qquad 1 < \frac{||z^{(L)}(x_{pert}) - z^{(L)}(x)||_2}{||z^{(L)}(x_{same}) - z^{(L)}(x)||_2}, \tag{8}$$

therefore,

$$\frac{||z^{(L)}(x_{same}) - z^{(L)}(x)||_2}{||x_{same} - x||_2} < \frac{||z^{(L)}(x_{pert}) - z^{(L)}(x)||_2}{||x_{pert} - x||_2}. \tag{9}$$

where x_{pert} is a data sample which is the results of an adversarial attack which is close to x but results in a different predicted label and x_{same} is another data sample that is close to x from the same class.

Using the mean value theorem [4], which states that for a $f(x)$ that is differentiable on the open interval (a,b) and continuous on the closed interval [a,b]. Then there is at least one point c in (a,b), such that:

$$f'(c) = \frac{f(b) - f(a)}{b - a} \tag{10}$$

According to the mean value theorem, let $[x, x_{pert}]$ be the D-dimensional line in the input space connecting x and x_{pert}, then there exists some $x' \in [x, x_{pert}]$ such that:

$$\frac{||z^{(L)}(x_{pert}) - z^{(L)}(x)||_2^2}{||x_{pert} - x||_2^2} \leq \sum_{k=1}^{K} ||\nabla_x z_k^{(L)}(x')||_2^2 = ||J(x')||_F^2. \tag{11}$$

This theory suggests that the ratio of the distance between the perturbed input x_{pert} and original input x with their class's output is upper bounded by the Frobenius norm of some x'. The lower the ratio, the more perturbations needed to change the class output. Hence a lower Frobenius norm of the network's Jacobian matrix will help the network to increase the robustness to small changes in the input space. The paper have demonstrated empirically that it leads to enhanced robustness with minimal change in the original network's accuracy and shown that the regularization can be applied as a post-processing method after regular training has finished.

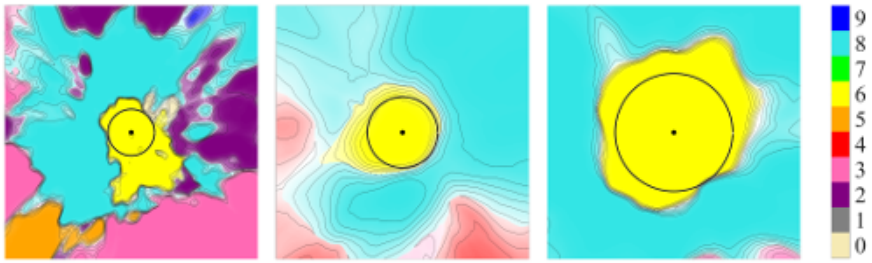
4 Experiments and Examples

The paper have provided experiment results which compares the average Frobenius norm of the Jacobian matrix at the original data without any defense method, and those with the Jacobian regularization defense method as shown in table 1. As expected, the average Frobenius norm on the perturbed input $x_{i_{pert}}$ is much larger for the network with no defense. This could results the network to classify the input as a different class from the original data.

Defense method	$\frac{1}{N}\sum_{i=1}^{N} ||J(x_i)||_F$	$\frac{1}{N}\sum_{i=1}^{N} ||J(x_{i_{pert}})||_F$
No defense	0.14	0.1877
Jacobian Regularization	0.0315	0.055

Table 1: Average Frobenius norm of the Jacobian matrix at the original data and the data perturbed by DeepFool. The DNN is trained on MNIST data

To further visualize the Jacobian Regularization, [5] output the class boundaries for the MNIST [6] data shown in figure 1. From the figure, we can observe that the class boundaries for network with Jacobian regularization has a larger radius which allows a larger amount of perturbations on the input to still fall into the same class compared to the ℓ_2 regularization.

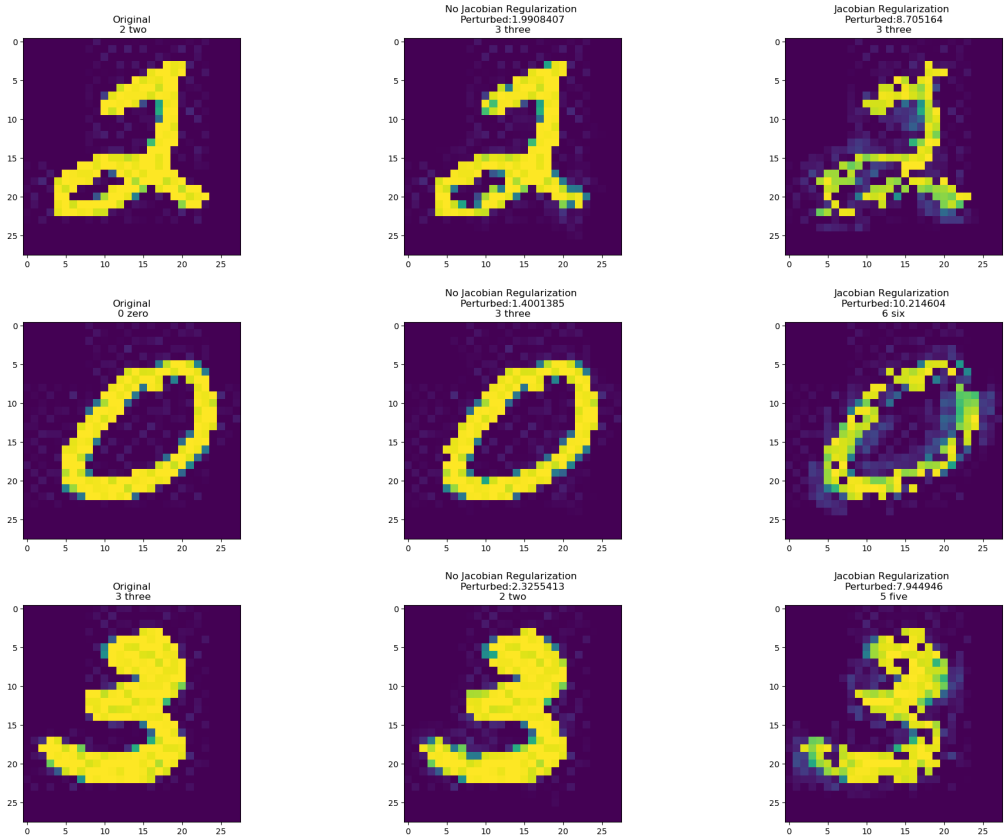


(a) Without regularization (b) With L^2 regularization (c) With Jacobian regularization

Figure 1: Class boundaries of MNIST data on LeNet architecture, different color represents the different classes. The black dot represents the original predicted data and the circle around it represent the minimum perturbation needed to change the data class.

The paper [3] uses the DeepFool method of finding the minimum perturbation $d*$ needed of an input x_i that results in the network to change the predicted class of x_i. For networks trained with either no regularization or only ℓ_2 regularization will have a smaller range of class decision boundaries. This would means that adversarial attacks will be able to change the predicted class with lesser perturbation for networks without Jacobian regularization. However, this also means that networks trained with the Jacobian regularization are not completely robust either. The adversarial attacks will just need to apply more perturbations to change the predicted class for those networks trained with the Jacobian regularization.

With more perturbations needed to tweak the predicted class of Jacobian regularization, the original input might have been changed to be non-recognizable visually. To visualize how much perturbations needed to change the predicted class, we perform a simple experiment to show the perturbed images in figure 2. We modify the codes [1] provided by [2] and [5] to conduct our experiments. The codes are also provided in the Supplementary Materials.



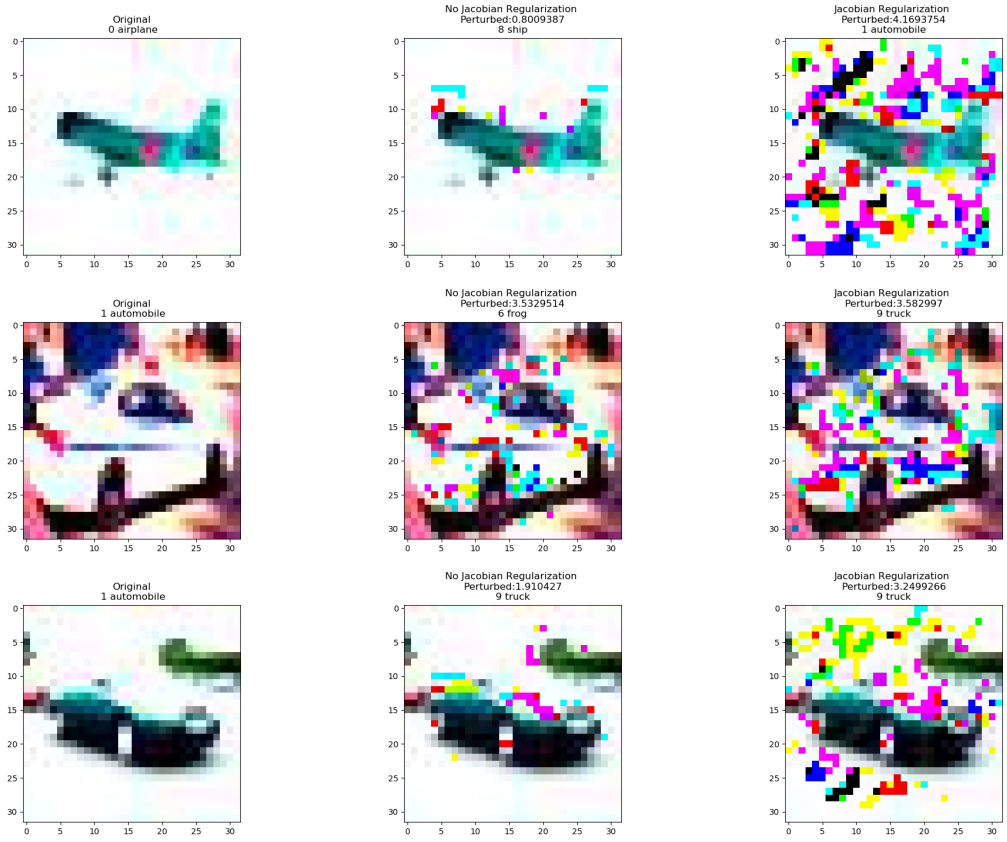
(a) (b) (c)

Figure 2: MNIST Dataset: (a) Original image of predicted class. (b) Perturbed image without Jacobian Regularization. (c) Perturbed image with Jacobian Regularization.

We can see from the experiment results that the amount of perturbation needed for changing the predicted class of the network trained with the Jacobian regularization is quite substantial. The amount of perturbations needed can be 7 times more than the network trained without using any Jacobian regularization for the number 0 case. This results in the distortion of the image that the hand-written stroke of the number has is almost unrecognizable. However, visually, we are still able to identify some of the numbers even with the perturbations. Even for simple MINST digits data, after the perturbations on Jacobian Regularization, we are still able to recognize most of the digits. It might not be so obvious to some that the images in (c) are perturbed.

To see how the Jacobian regularization works for more complex data, we decided to try out more experiments with CIFAR10 [7] data that consists of real life images with RGB channel. The results are shown in figure 3 below:

[1] `https://github.com/jxlim89/CS5339Project`



(a) (b) (c)

Figure 3: CIFAR10 Dataset: (a) Original image of predicted class. (b) Perturbed image without Jacobian Regularization. (c) Perturbed image with Jacobian Regularization.

For RGB channels, the perturbed images are more obvious as the DeepFool algorithm is able to add perturbations to more channels, which we believe is able to affect the decision boundary class easier with significantly lesser perturbations than MNIST data set. From this experiment, we could see that the perturbed image of RGB images are more affected visually due to the unnatural coloring. We could interpret that the Jacobian Regularization would work more effectively on RGB images as the perturbed images are more affected visually with the added number of channels. In addition, it is now more obvious than MNIST data set that images in (c) has been perturbed.

5 Discussion and Conclusion

In conclusion, we are able to understand the theory behind how the Jacobian regularization makes the network more robust and able to answer our question:*Are Deep Neural Networks completely robust by using Jacobian regularization?* Which is unfortunately not, however, the amount of perturbation needed to trick the network that has been trained with the Jacobian regularization does increase quite a substantial amount compared to using normal ℓ_2 regularization. With current state-of-the-art adversarial attacks that exploit the network class decision boundaries, networks are able to be tricked easily using simple calculations stated in [2]. We believed in order to make the network more robust, we cannot rely only on increasing the class decision boundaries, the features of the image that the DNN models learned have to be more meaningful in order to tell the difference between perturbed images and original image. In this note, future works may take into consideration of using GAN models [8] to train the model to learn what features are important. As we believed that the discriminator of a state-of-the-art GAN model can be trained to distinguish important features from the generator.

References

[1] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2014.

[2] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. *CoRR*, abs/1511.04599, 2015.

[3] Daniel Jakubovitz and Raja Giryes. Improving DNN robustness to adversarial attacks using jacobian regularization. *CoRR*, abs/1803.08680, 2018.

[4] Weisstein and Eric W. Mean-value theorem from mathworld–a wolfram web resource. `https://mathworld.wolfram.com/Mean-ValueTheorem.html`.

[5] Judy Hoffman, Daniel A. Roberts, and Sho Yaida. Robust learning with jacobian regularization. *ArXiv*, abs/1908.02729, 2019.

[6] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[7] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[8] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial networks. *ArXiv*, abs/1406.2661, 2014.