Certified Defense to Adversarial Attack with Nonexpansive Neural Networks

Github Repository

Binjie Yuan[1], Xuguang Duan[1], Feiyu Xiao[2]

[1]Department of Computer Science and technology

[2]School of Vehicle and Mobility

Introduction

Deep neural networks are known to be vulnerable to adversarial examples. A myriad of defenses methods have been proposed to counter this problem, most of which are soon broken by stronger attacks. Certified defense methods that provide guaranteed robustness to norm-bounded attacks have been invented as a promising approach to end this arm race of AI security. Recently, nonexpansive neural network has emerged as a scalable way to obtain neural networks with certified robustness[1], and we reproduce and further improve this work.

This work aims to obtain guaranteed robustness through constraining the global Lipschitz constant of neural networks[2] and the following three conditions are combined to enhance robustness:

1. the Lipschitz constant of a network from inputs to logits is no greater than 1 with respect to the L2-norm;
2. the loss function explicitly maximizes confidence gap, which is the difference between the largest and second largest logits of a classifier;
3. the network architecture restricts confidence gaps as little as possible.

L2-nonexpansive neutral network

Normal Pooling

Max-pooling is replaced with norm-pooling, which was reported to occasionally increase accuracy[3]. Instead of taking the max of values inside a pooling window, we take the L2-norm of them. It is straightforward to verify that norm-pooling is nonexpansive and would entirely preserve the L_2-distance, which ensure its effectiveness in increasing confidence gaps.

If pooling windows overlap, we divide the input tensor by $\sqrt{K}$ where K is the maximum number of pooling windows in which an entry can appear.

Two sided ReLU

We use two-sided **ReLU** which is a function from R to R^2 and simply computes **ReLU**(x) and **ReLU**(x), which has been studied for accuracy improvement in convolution layers[4]. Two-sided ReLU is nonexpansive with respect to any L_p-norm and can overcome the general ReLU distance loss problems.

Loss function

A loss function with three terms, with trade-off hyperparameters γ and ω is used:

$$\mathcal{L} = \mathcal{L}_a + \gamma \cdot \mathcal{L}_b + \omega \cdot \mathcal{L}_c$$

Let $y_1, y_2, \cdots, y_K$ be outputs from classifier and $u_1, u_2, \cdots, u_K$ be trainable parameters, and the first and second loss items are:

$$\mathcal{L}_a = \text{softmax-cross-entropy}\left(u_1 y_1, \cdots, u_K y_K, \text{label}\right)$$

$$\mathcal{L}_b = \text{softmax-cross-entropy}\left(v y_1, \cdots, v y_K, \text{label}\right)$$

where v can be either a trainable parameter or a hyperparameter.Below is the thrid loss term with z a hyperparameter:

$$\mathcal{L}_c = \frac{\operatorname{ave}\left(\log\left(1 - \operatorname{softmax}\left(z y_1, \cdots, z y_K\right)_{\text{label}}\right)\right)}{z}$$

Experimental Settings

We perform experiments on the widely used MNIST dataset.
To evaluate the models' performance under adversarial attack, we use the FGSM and PGD attack method implemented in *cleverhans* library.

Specific Settings

	eps	clip_{min}	clip_{max}	eps_{iter}	nb_{iter}
FGSM	0.3	0.0	1.0		
PGD	0.3	0.0	1.0	0.01	40

Extensive design choice of L2NNN

We extensively experimented with some design choices of L2NNN.

Table 1: Accuracies of MNIST classifiers under different attacks

Model Parameters	NAT	FGSM	PGD	c gap[a]
w:0.0 dbc[b]:False	86.89%	86.90%	86.10%	$4.796e^{-3}$
w:0.0 dbc:True	82.30%	82.25%	70.59%	$5.533e^{-3}$
w:0.1 dbc:False	88.83%	88.42%	65.35%	$3.372e^{-3}$
w:0.1 dbc:True	99.00%	3.66%	0.91%	$7.211e^{-1}$
w:0.5 dbc:False	79.37%	77.05%	65.85%	$7.950e^{-4}$
w:0.5 dbc:True	97.42%	4.32%	3.10%	$5.776e^{-1}$

[a] Condidence Gap

[b] Divide before convolution

Extend L2NNN to ResNet

We also extend the L2NNN framework to modern neural network architectures like ResNet.

Table 2: Results with ResNet added

Model Parameters	NAT	FGSM	PGD	c gap
w:0.0 dbc:False	66.94%	66.91%	57.81%	1.642
w:0.0 dbc:True	20.37%	20.32%	22.33%	1.116
w:0.1 dbc:False	75.05%	73.90%	61.39%	9.103
w:0.1 dbc:True	98.83%	1.16%	0.34%	1.678
w:0.5 dbc:False	90.44%	90.21%	54.79%	2.230
w:0.5 dbc:True	93.92%	3.99%	1.07%	0.470

Summary and conclusions

We have reproduced and modified L2-nonexpansive neural networks of the work[1], and we also fixed the errors in the paper and made further improvements to the network.

Our main contributions are:

- We extensively experimented with some design choices of L2NNN.
- We extend the L2NNN framework to modern neural network architectures like ResNet.

Our main findings are:

- We find that L2-non-expansive ResNet can achieve larger confidence gap, thus provides more certified defense against white box adversarial attacks.
- We find that although some design options of L2NNN (like adding weight regularization loss and divide kernel size before convolution) can achieve better confidence gap (thus promoting certified defense), downgrade the model's performance under real attacks like FGSM and PGD. Since attacking algorithms give an upper bound of the model's performance while confidence gaps give a lower bound, the tradeoff between these two bounds is a thought-provoking problem.

References

[1] Haifeng Qian and Mark N. Wegman, L2-Nonexpansive Neural Networks, *ICLR.2019*

[2] Tsuzuku, Yusuke,Sato, Issei,Sugiyama, Masashi, Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks, *NeurIPS.2018*

[3] Boureau Y L, Ponce J, LeCun Y., A theoretical analysis of feature pooling in visual recognition, *ICML 2010*

[4] Shang W , Sohn K , Almeida D , et al., Understanding and Improving Convolutional Neural Networks via Concatenated Rectified Linear Units, *ICML 2016*

Final Presentation for ML Course, June, 2019, Beijing