Adversarial Robustness of Stabilized Neural ODE Might be from Obfuscated Gradients

Yifei Huang[1], Yaodong Yu[2], Hongyang Zhang[3], Yi Ma[2], and Yuan Yao[1]

[1]Department of Mathematics, Hong Kong University of Science and Technology
[2]Department of EECS, University of California at Berkeley
[3]University of Waterloo and Toyota Technological Institute at Chicago
Email: hongyanz@ttic.edu, yuany@ust.hk

Outline

- Adversarial Examples
- Adversarial Attacks and Defences
- Neural Ordinary Differentials Equations
- Deep Stable ODE Network
- Experiments
  - Positive Experiments
  - Negative Experiments
  - Adaptive Stepsize Analysis
- Summary

Adversarial Examples

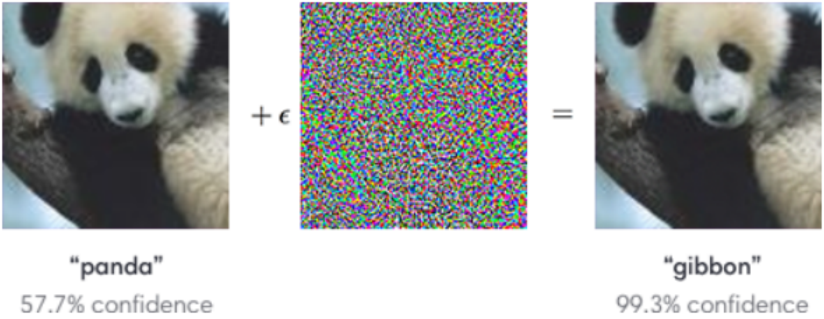


Figure: An adversarial input imperceivable to human, overlaid on a typical image, can cause a classifier to misclassify a panda as a gibbon.[**Goodfellow et al. (2014)**]

Adversarial Examples

Formal Definition [Szegedy et al. (2013)]: adversarial examples are generated by minimizing the following function with respect to r

$$\mathcal{L}(\hat{f}(x+r), l) + c \cdot |r| \text{ subject to } x + r \in [0,1] \quad (1)$$

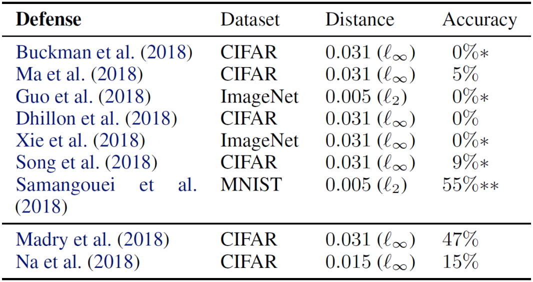
where $\mathcal{L}$ is the loss function, such as cross entropy loss, $\hat{f}$ is the model we want to attack, x is the original image, l is the target label and r is the pixel-wise perturbation, then $x_{adv} = x + r$.

Adversarial Attacks

- Gradient Based
  - Fast Gradient Sign Method (FGSM)
  - Projected Gradient Descent (PGD)
  - Carlini & Wagner attack (C&W)
  - ...
- Gradient Free
  - SPSA attack
  - Boundary attack
  - ...

Obfuscated Gradients

Gradient masking (Papernot et al. (2017); Athalye et al. (2018)) is a phenomenon widely associated with the obfuscation of gradient information in gradient based adversarial attacks, yet failure under robust gradient and gradient-free attacks, thus giving a false sense of adversarial robustness.



Defense	Dataset	Distance	Accuracy
Buckman et al. (2018)	CIFAR	0.031 (ℓ_∞)	0%*
Ma et al. (2018)	CIFAR	0.031 (ℓ_∞)	5%
Guo et al. (2018)	ImageNet	0.005 (ℓ_2)	0%*
Dhillon et al. (2018)	CIFAR	0.031 (ℓ_∞)	0%
Xie et al. (2018)	ImageNet	0.031 (ℓ_∞)	0%*
Song et al. (2018)	CIFAR	0.031 (ℓ_∞)	9%*
Samangouei et al. (2018)	MNIST	0.005 (ℓ_2)	55%**
Madry et al. (2018)	CIFAR	0.031 (ℓ_∞)	47%
Na et al. (2018)	CIFAR	0.015 (ℓ_∞)	15%

Figure: Seven of nine defense techniques accepted at ICLR 2018 cause obfuscated gradients and are vulnerable to their attacks. Athalye et al. (2018)

Adversarial Defenses

- Adversarial Training
    - Generating adversarial examples and include them as part of the training data. (**Madry et al. (2017)**; **Zhang et al. (2019)**)
- Random Smoothing and stability training
    - Adding gaussian noise to the original images. (**Cohen et al. (2019)**, **Zheng et al. (2016)**)
- ...

Adversarial Training

Adversarial training is first introduced by [**Madry et al. (2017)**] which tries to solve the following minimax problem:

$$\min_{\theta} \rho(\theta), \quad \text{where} \quad \rho(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\delta \in \mathcal{S}} L(\theta, x+\delta, y) \right] \tag{2}$$

where $\mathbb{E}_{\mathcal{D}}[L]$ is the population risk for data distribution $\mathcal{D}$. Instead of feeding original samples to loss L, we allow the adversary to perturb the input first, thus making model to gain adversarial robustness.

Adversarial Training

Problems

- Norm-agnostic Setting
  - Adversarial training suffers from brittleness against attacks in ℓ_2 and ℓ_∞ norms simultaneously. (**Li et al. (2019)**)
- Intrinsic trade-off between natural accuracy and adversarial robustness
  - Adversarial training typically leads to more than 10% reduction of accuracy compared with natural training. (**Zhang et al. (2019)**; **Tsipras et al. (2018)**)
- Computationally very expensive.

Adversarial Training

Problems

- Norm-agnostic Setting
  - Adversarial training suffers from brittleness against attacks in ℓ_2 and ℓ_∞ norms simultaneously. (**Li et al. (2019)**)
- Intrinsic trade-off between natural accuracy and adversarial robustness
  - Adversarial training typically leads to more than 10% reduction of accuracy compared with natural training. (**Zhang et al. (2019)**; **Tsipras et al. (2018)**)
- Computationally very expensive.

Questions: Instead of adversarial training, can we design a stable network architecture whose natural training is able to gain adaptive robustness on both ℓ_2 and ℓ_∞ norms while no sacrifice of natural accuracy?

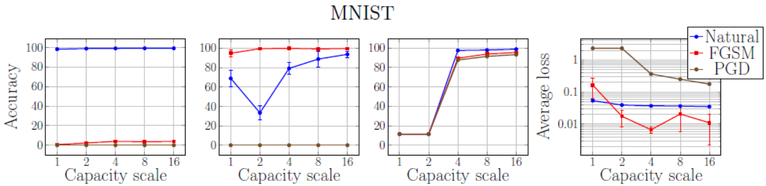
Adversarial Training

Previous works on robust network architecture design

- Parseval networks (**Cisse et al. (2017)**): explicitly bounded the Lipschitz constant by either requiring each fully-connected or convolutional layer be composed of orthonormal filters.
    - Pros: control Lipschitz constants of neural networks through regularization.
    - Cons: the robustness is much weaker than adversarial training.
- ℓ_2-nonexpansive neural networks (**Qian and Wegman (2018)**): restricting the spectral radius of the matrix in each layer to be small.
    - Pros: control Lipschitz constants of neural networks without requiring filtes to be orthogonal to each other.
    - Cons: can not guarantee robustness in ℓ_∞ norm.
- ...

Intuitions with deep neural networks

[**Madry et al. (2017)**] empirically reported that the capacity of the model becomes a major factor affecting its overall robustness. Inspired by this, What happens as the network goes deeper and takes smaller steps? This naturally leads to the ODE network. Since ODE networks are provable deep limit of ResNets (**Avelin and Nyström (2019)**;**Thorpe and van Gennip (2018)**)



CIFAR10

	Simple	Wide
Natural	92.7%	95.2%
FGSM	27.5%	32.7%
PGD	0.8%	3.5%

(a) Natural training

Simple	Wide
87.4%	90.3%
90.9%	95.1%
0.0%	0.0%

(b) FGSM training

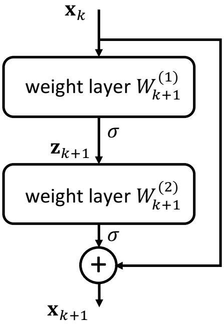
Simple	Wide
79.4%	87.3%
51.7%	56.1%
43.7%	45.8%

(c) PGD training

Simple	Wide
0.00357	0.00371
0.0115	0.00557
1.11	0.0218

(d) Training Loss

Neural Ordinary Differentials Equations



(a) ResNet block

$$\frac{\mathbf{z}_{k+1}-\mathbf{z}_k}{\Delta t}=\sigma(\mathbf{W}_{k+1}^{(1)}\mathbf{x}_k),$$
$$\frac{\mathbf{x}_{k+1}-\mathbf{x}_k}{\Delta t}=\sigma(\mathbf{W}_{k+1}^{(2)}\mathbf{z}_{k+1}),$$
$$\mathbf{z}_k=\mathbf{0},\ \Delta t=1,$$

(b) ResNet equation

Figure: ResNet. [**He et al. (2016)**]

Neural Ordinary Differentials Equations

In the Neural ODE, in constrast, [**Chen et al. (2018)**] took the limit of the finite differences over the infinitesimal Δt and parameterized the continuous dynamics of hidden units using an ODE specified by a neural network.

$$
\begin{aligned}
\frac{\mathbf{z}_{k+1}-\mathbf{z}_k}{\Delta t} &= \sigma(\mathbf{W}^{(1)}_{k+1}\mathbf{x}_k),\\
\frac{\mathbf{x}_{k+1}-\mathbf{x}_k}{\Delta t} &= \sigma(\mathbf{W}^{(2)}_{k+1}\mathbf{z}_{k+1}),\\
\mathbf{z}_k = \mathbf{0},\ \Delta t &= 1,
\end{aligned}
$$

(a) ResNet equation

$$
\begin{aligned}
\frac{d\mathbf{x}(t)}{dt} &= \sigma(\mathbf{W}^{(2)}_{k+1}\sigma(\mathbf{W}^{(1)}_{k+1}\mathbf{x}(t))),\\
\mathbf{x}_{k+1} &= \mathbf{x}(t_0), \quad \mathbf{x}(0) = \mathbf{x}_k,
\end{aligned}
$$

(b) ResNet ODE

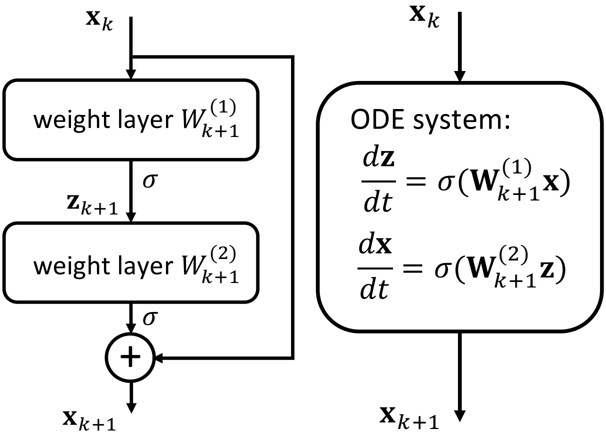
Figure: ResNet Equation vs ODE. [**Chen et al. (2018)**]

Deep Stable ODE Networks

Inspired by Neural ODE, we introduce the following parametric ODE block with a small positive damping factor γ

$$
\begin{aligned}
\frac{d\mathrm{x}(t)}{dt} &= \sigma(\mathrm{W}^{(2)}_{k+1}\mathrm{z}(t) - \gamma\mathrm{x}),\\
\frac{d\mathrm{z}(t)}{dt} &= \sigma(\mathrm{W}^{(1)}_{k+1}\mathrm{x}(t) - \gamma\mathrm{z}),\\
\mathrm{x}_{k+1} &= \mathrm{z}(t_0), \quad \mathrm{x}(0) = \mathrm{x}_k, \quad \mathrm{z}(0) = \mathrm{z}_k,
\end{aligned}
\tag{3}
$$

Deep Stable ODE Networks



ODE system:

$$\frac{d\mathbf{z}}{dt} = \sigma(\mathbf{W}^{(1)}_{k+1}\mathbf{x})$$

$$\frac{d\mathbf{x}}{dt} = \sigma(\mathbf{W}^{(2)}_{k+1}\mathbf{z})$$

(a) ResNet block

(b) Our Stable ODE block

Figure: ODE block architecture.

Deep Stable ODE Networks

Theorem (Stability of ODE Blocks)

Suppose that the activation function σ is strictly monotonically increasing, i.e., $\sigma'(\cdot) > 0$ and positive damping factor γ is small. Let $\mathsf{W}_{k+1}^{(2)} = -\mathsf{W}_{k+1}^{(1)\top}$. Then for any implementation of network parameters, the forward propagation (3) is stable in the sense of Lyapunov; that is, for all $\delta > 0$, there exists a stable radius $\epsilon(\delta) > 0$ such that if $\|\mathsf{x}_0 - \mathsf{x}_0'\| \leq \epsilon(\delta)$, we have $\|f_{\text{ODENet-k}(\mathsf{x}_0;t_0)} - f_{\text{ODENet-k}(\mathsf{x}_0';t_0)}\| \leq \delta$ for all $t_0 > 0$.

$$\frac{d}{dt}\begin{bmatrix}\mathbf{x}\\ \mathbf{z}\end{bmatrix} = \sigma\left(\begin{bmatrix}\mathbf{0} & -\mathbf{W}_{k+1}^{\top}\\ \mathbf{W}_{k+1} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \mathbf{z}\end{bmatrix} - \gamma\mathbf{I}\begin{bmatrix}\mathbf{x}\\ \mathbf{z}\end{bmatrix}\right),$$
$$\mathbf{x}(0) = \mathbf{x}_k,\ \mathbf{z}(0) = \mathbf{z}_k,\ \mathbf{x}_{k+1} := \mathbf{z}(t_0).$$

Figure: Skew-symmetric ODE Block

Deep Stable ODE Networks

Benefits of our skew-symmetric architecture:

- **Change of dimensionality**: the introduction of the auxiliary variable $z \in \mathbb{R}^{d_{out}}$ enables us to change the dimension of the input and output vectors; that is, the input variable $x \in \mathbb{R}^{d_{in}}$ may have different dimensions as the output variable $z \in \mathbb{R}^{d_{out}}$. This is in sharp contrast to the original design of Neural ODE (**Chen et al. (2018)**), where the input and output vectors of each ODE block must have the same dimension.
- **Parameter efficiency**: the skew-symmetric ODE block has only half number of parameters compared to the ResNet blocks and the original design of Neural ODE blocks due to parameter sharing.
- **Inference-time robustness**: the established architecture enjoys stability.

Deep Stable ODE Networks

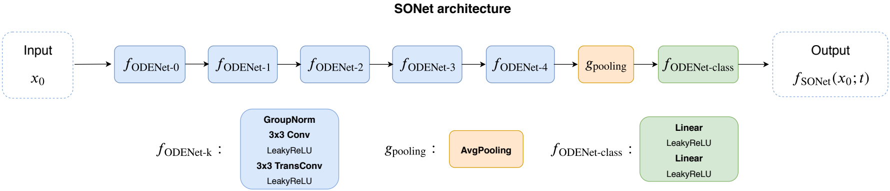


Figure: Stabilized neural ODE Network (SONet) architecture example. Both $f_{\text{ODENet-k}}$ and $f_{\text{ODENet-class}}$ are built on our stable ODE block.

Experiments

Models

- **ResNet**: We apply the ResNet with 10 layers as the baseline model, denoted by ResNet10.
- **SONet**: We replace each residual basic block in the ResNet10 architecture with the proposed stable skew-symmetric ODE block.
- **SOBlock**: We replace the first convolution layer in ResNet10 above by our proposed skew-symmetric ODE block and leave the other parts unchanged.
- **ODENet**: We use the original ODENet architecture (**Chen et al. (2018)**).
- Additionally, in order to compare the performance of SONet and ResNet with different number of parameters, we scale the model capacity of SONet and ResNet10 by changing the input channel from 32 to 64.

Experiments

Attacks

- **White-Box Attacks**
  - ℓ_∞ PGD attack: we set the perturbation distance ϵ= 0.031 and the attack step size α=0.003.
  - ℓ_2 PGD attack: we set the perturbation distance ϵ= 0.5 and the attack step size α=0.1.
  - ℓ_∞ CW attack: we set the perturbation distance $\epsilon = 0.031$, the max-iterations K=100.
- **Black-Box Attack**
  - SPSA attack: we apply the $\epsilon = 0.031$, the number of iterations K=20 and the number of samples to be 32.

Training Method

- **Adversarial training** : We use TRADES [Zhang et al. (2019)] as our baseline adversarial training method with two different regularization parameter $1/\lambda = 1.0$ and 6.0.

Experiments

ODE Solvers (we set all error tolerance as 0.1):

- Fixed Stepsize:
  - Euler method (first order, fixed step size $h = 1$)
  - RK4 (fourth order, fixed step size $h = 1$)
- Adaptive Stepsize:
  - Heun (second order, adaptive step size)
  - Bosh3 (third order, adaptive stepsize)
  - DOPRI5 (fifth order, adaptive step size)

Training settings: We set the total epoch T = 350, batch size B = 100, the initial learning rate $\eta = 0.01$ (decay 0.1 at 150 and 300 epochs respectively), and apply SGD with momentum 0.9 as the optimizer. No weight decay is used during training.

Positive Experiments - PGD Attacks

Table 1: Comparisons between SONet, SOBlock with natural training and ResNet10 with TRADES under white-box PGD adversarial attacks on CIFAR10 dataset.

Model	Channel	Under which attack	$\mathcal{A}_{\mathrm{nat}}(f)$	$\mathcal{A}_{\mathrm{rob}}(f)$	
				$\epsilon = 0.031(\ell_\infty)$	$\epsilon = 0.5(\ell_2)$
SONet	32	PGD^{20}	88.08%	53.67%	57.39%
SOBlock	32	PGD^{20}	90.28%	58.21%	60.25%
ResNet10-TRADES ($1/\lambda = 1.0$)	32	PGD^{20}	81.52%	35.26%	57.07%
ResNet10-TRADES ($1/\lambda = 6.0$)	32	PGD^{20}	73.69%	43.46%	55.73%
SONet	64	PGD^{20}	89.36%	61.62%	64.08%
SOBlock	64	PGD^{20}	**91.57%**	**62.35%**	**64.70%**
ResNet10-TRADES ($1/\lambda = 1.0$)	64	PGD^{20}	82.74%	37.64%	58.97%
ResNet10-TRADES ($1/\lambda = 6.0$)	64	PGD^{20}	76.29%	45.24%	57.28%
SONet	32	PGD^{1,000}	88.08%	19.62%	31.75%
SOBlock	32	PGD^{1,000}	90.28%	52.01%	52.79%
ResNet10-TRADES ($1/\lambda = 1.0$)	32	PGD^{1,000}	81.52%	33.60%	56.70%
ResNet10-TRADES ($1/\lambda = 6.0$)	32	PGD^{1,000}	73.69%	43.30%	55.48%
SONet	64	PGD^{1,000}	89.36%	24.25%	39.79%
SOBlock	64	PGD^{1,000}	**91.57%**	**55.43%**	57.37%
ResNet10-TRADES ($1/\lambda = 1.0$)	64	PGD^{1,000}	82.74%	35.78%	**58.73%**
ResNet10-TRADES ($1/\lambda = 6.0$)	64	PGD^{1,000}	76.29%	44.70%	56.87%

Negative Experiments - CW&SPSA Attacks

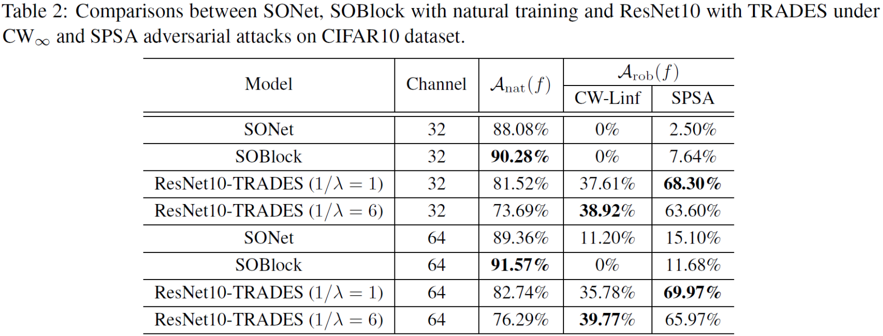
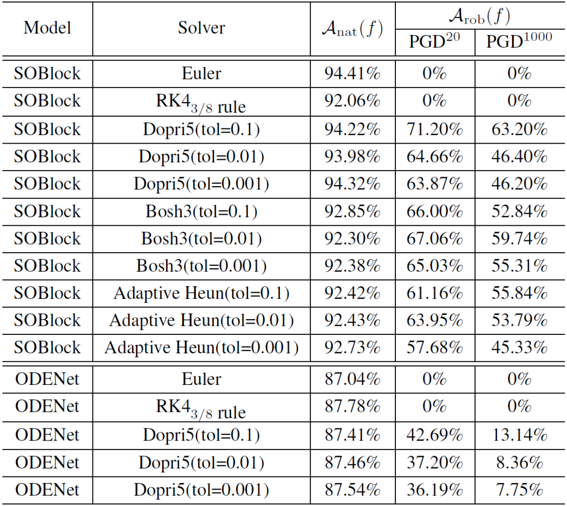


Table 2: Comparisons between SONet, SOBlock with natural training and ResNet10 with TRADES under CW_∞ and SPSA adversarial attacks on CIFAR10 dataset.

Model	Channel	$\mathcal{A}_{\mathrm{nat}}(f)$	$\mathcal{A}_{\mathrm{rob}}(f)$	
			CW-Linf	SPSA
SONet	32	88.08%	0%	2.50%
SOBlock	32	**90.28%**	0%	7.64%
ResNet10-TRADES ($1/\lambda = 1$)	32	81.52%	37.61%	**68.30%**
ResNet10-TRADES ($1/\lambda = 6$)	32	73.69%	**38.92%**	63.60%
SONet	64	89.36%	11.20%	15.10%
SOBlock	64	**91.57%**	0%	11.68%
ResNet10-TRADES ($1/\lambda = 1$)	64	82.74%	35.78%	**69.97%**
ResNet10-TRADES ($1/\lambda = 6$)	64	76.29%	**39.77%**	65.97%

Experiments - Adaptive Stepsize Analysis

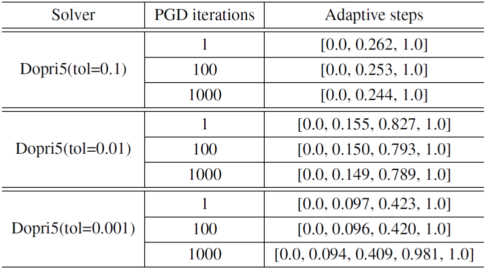
Table 3: Comparisons between SOBlock and ODENet with different solver ODE solvers under PGD adversarial attacks on CIFAR10 dataset.



Model	Solver	$\mathcal{A}_{\text{nat}}(f)$	$\mathcal{A}_{\text{rob}}(f)$	
			PGD^{20}	PGD^{1000}
SOBlock	Euler	94.41%	0%	0%
SOBlock	RK4$_{3/8}$ rule	92.06%	0%	0%
SOBlock	Dopri5(tol=0.1)	94.22%	71.20%	63.20%
SOBlock	Dopri5(tol=0.01)	93.98%	64.66%	46.40%
SOBlock	Dopri5(tol=0.001)	94.32%	63.87%	46.20%
SOBlock	Bosh3(tol=0.1)	92.85%	66.00%	52.84%
SOBlock	Bosh3(tol=0.01)	92.30%	67.06%	59.74%
SOBlock	Bosh3(tol=0.001)	92.38%	65.03%	55.31%
SOBlock	Adaptive Heun(tol=0.1)	92.42%	61.16%	55.84%
SOBlock	Adaptive Heun(tol=0.01)	92.43%	63.95%	53.79%
SOBlock	Adaptive Heun(tol=0.001)	92.73%	57.68%	45.33%
ODENet	Euler	87.04%	0%	0%
ODENet	RK4$_{3/8}$ rule	87.78%	0%	0%
ODENet	Dopri5(tol=0.1)	87.41%	42.69%	13.14%
ODENet	Dopri5(tol=0.01)	87.46%	37.20%	8.36%
ODENet	Dopri5(tol=0.001)	87.54%	36.19%	7.75%

Experiments - Adaptive Stepsize Analysis

Table 4: Adaptive steps with ODENet under different dopri5 tolerances and PGD$_\infty$ attack iterations



Solver	PGD iterations	Adaptive steps
Dopri5(tol=0.1)	1	[0.0, 0.262, 1.0]
	100	[0.0, 0.253, 1.0]
	1000	[0.0, 0.244, 1.0]
Dopri5(tol=0.01)	1	[0.0, 0.155, 0.827, 1.0]
	100	[0.0, 0.150, 0.793, 1.0]
	1000	[0.0, 0.149, 0.789, 1.0]
Dopri5(tol=0.001)	1	[0.0, 0.097, 0.423, 1.0]
	100	[0.0, 0.096, 0.420, 1.0]
	1000	[0.0, 0.094, 0.409, 0.981, 1.0]

With fixed solver, different PGD iterations will result in different adaptive steps, thus confounding the gradients for PGD attacks. On the other hand, for fixed PGD iterations 1000, with the decrease of error tolerance, the selected adaptive stepsize becomes smaller and smaller which means that the estimated solution is more accuracy and gradient masking effect is decreasing.

Summary

- We design a stabilized neural ODE network named SONet whose ODE blocks are skew-symmetric and proved to be input-output stable.

Summary

- We design a stabilized neural ODE network named SONet whose ODE blocks are skew-symmetric and proved to be input-output stable.
- By experiments, we show that SONet with only natural training can achieve comparable robustness with the state-of-the-art adversarial defense methods under PGD attacks, without sacrificing natural accuracy.

Summary

- We design a stabilized neural ODE network named SONet whose ODE blocks are skew-symmetric and proved to be input-output stable.
- By experiments, we show that SONet with only natural training can achieve comparable robustness with the state-of-the-art adversarial defense methods under PGD attacks, without sacrificing natural accuracy.
- Moreover, we find that the adaptive stepsize numerical ODE solvers, such as adaptive HEUN2, BOSH3, and DOPRI5, have a gradient masking effect that fails the PGD attacks which are sensitive to gradient information of training loss.
- We provide a new explanation that the adversarial robustness of ODE-based networks mainly comes from the obfuscated gradients in numerical ODE solvers with adaptive step sizes.

Appendix

Sketch of Proof.

We observe that Eqn. (3) has an equivalent expression, $\mathrm{x}_{k+1} = f_{\mathrm{ODENet}}(\mathrm{x}_k; t_0)$:

$$\frac{d}{dt}\begin{bmatrix}\mathrm{x}\\ \mathrm{z}\end{bmatrix} = \sigma\left(\begin{bmatrix}0 & -\mathrm{W}_{k+1}\\ \mathrm{W}_{k+1}^\top & 0\end{bmatrix}\begin{bmatrix}\mathrm{x}\\ \mathrm{z}\end{bmatrix} - \gamma\mathrm{I}\begin{bmatrix}\mathrm{x}\\ \mathrm{z}\end{bmatrix}\right),$$

$$\mathrm{x}(0) = \mathrm{x}_k,\ \mathrm{z}(0) = \mathrm{z}_k,\ \mathrm{x}_{k+1} := \mathrm{z}(t_0).$$

Denote by

$$\mathrm{A}_{k+1} := \begin{bmatrix}0 & -\mathrm{W}_{k+1}^\top\\ \mathrm{W}_{k+1} & 0\end{bmatrix}.$$

Note that A_{k+1} is a skew-symmetric matrix such that $\mathrm{A}_{k+1} = -\mathrm{A}_{k+1}^\top$. So $Re[\lambda_i(\mathrm{A}_{k+1})] \leq 0$ for all i, where $Re[\cdot]$ represents the real part of a complex variable and $\lambda_i(\mathrm{A}_{k+1})$ is the i-th eigenvalue of matrix A_{k+1}. □

Sketch of proof.

We note that an ODE system is stable if $Re[\lambda_i(\mathsf{J}_{k+1})] < 0$ (**Åström and Murray (2010)**), where J_{k+1} is the Jacobian of the ODE:

$$\mathsf{J}_{k+1} := \nabla_{[\mathsf{x};\mathsf{z}]}\left(\sigma\left(\begin{bmatrix} 0 & -\mathsf{W}_{k+1}^\top \\ \mathsf{W}_{k+1} & 0 \end{bmatrix}\begin{bmatrix}\mathsf{x}\\ \mathsf{z}\end{bmatrix} - \gamma \mathsf{I}\begin{bmatrix}\mathsf{x}\\ \mathsf{z}\end{bmatrix}\right)\right)$$
$$=: \mathsf{D}_{k+1}(\mathsf{A}_{k+1} - \gamma \mathsf{I}),$$

where we have defined

$$\mathsf{D}_{k+1} := \mathrm{Diag}\left(\sigma'\left(\begin{bmatrix} -\gamma & -\mathsf{W}_{k+1}^\top \\ \mathsf{W}_{k+1} & -\gamma \end{bmatrix}\begin{bmatrix}\mathsf{x}\\ \mathsf{z}\end{bmatrix}\right)\right).$$

Because $\sigma'(\cdot) > 0$, the matrix $\mathsf{D}_{k+1}^{-1/2}$ exists. We observe that

$$\mathsf{J}_{k+1} \sim \mathsf{D}_{k+1}^{1/2}(\mathsf{A}_{k+1} - \gamma \mathsf{I})\mathsf{D}_{k+1}^{1/2},$$

where the notation $\sim$ means the two matrices are similar. □

Appendix

Sketch of Proof.

Since similar matrices have the same eigenvalues, for all i, we have

$$\lambda_i(\mathrm{J}_{k+1}) = \lambda_i(\mathrm{D}_{k+1}^{1/2}(\mathrm{A}_{k+1} - \gamma \mathrm{I})\mathrm{D}_{k+1}^{1/2}). \tag{4}$$

For the right hand side in Eqn. (4), $Re[\lambda_i(\mathrm{A}_{k+1})] \leq 0$ So $Re[\lambda_i(\mathrm{A}_{k+1} - \gamma \mathrm{I})] < 0$, and matrix D_{k+1} is positive diagonal. Combining with Eqn. (4), we have $Re[\lambda_i(\mathrm{J}_{k+1})] < 0$. Thus, the Lyapunov stability is valid with respect to Euclidean ℓ_2-norm. For other equivalent ℓ_p-norms $(1 \leq p \leq \infty)$, the result holds up to a constant that depends on the input dimension. The proof is completed.

□

References I

Karl Johan Åström and Richard M Murray. *Feedback systems: an introduction for scientists and engineers*. Princeton university press, 2010.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, 2018.

Benny Avelin and Kaj Nyström. Neural odes as the deep limit of resnets with constant weights. *arXiv preprint arXiv:1906.12183*, 2019.

Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in neural information processing systems*, pages 6571–6583, 2018.

References II

Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 854–863. JMLR. org, 2017.

Jeremy M Cohen, Elan Rosenfeld, and J Zico Kolter. Certified adversarial robustness via randomized smoothing. *arXiv preprint arXiv:1902.02918*, 2019.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

References III

Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. On norm-agnostic robustness of adversarial training. *arXiv preprint arXiv:1905.06455*, 2019.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519, 2017.

Haifeng Qian and Mark N Wegman. L2-nonexpansive neural networks. *arXiv preprint arXiv:1802.07896*, 2018.

References IV

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Matthew Thorpe and Yves van Gennip. Deep limits of residual neural networks. *arXiv preprint arXiv:1810.11741*, 2018.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan. Theoretically principled trade-off between robustness and accuracy. *arXiv preprint arXiv:1901.08573*, 2019.

Tianyuan Zhang and Zhanxing Zhu. Interpreting adversarially trained convolutional neural networks. *arXiv preprint arXiv:1905.09797*, 2019.

References V

Stephan Zheng, Yang Song, Thomas Leung, and Ian Goodfellow. Improving the robustness of deep neural networks via stability training. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 4480–4488, 2016.