TOWARDS NATURAL ROBUSTNESS AGAINST ADVERSARIAL EXAMPLES

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent studies have shown that deep neural networks are vulnerable to adversarial examples, but most of the methods proposed to defense adversarial examples cannot solve this problem fundamentally. In this paper, we theoretically prove that there is an upper bound for neural networks with identity mappings to constrain the error caused by adversarial noises. However, in actual computations, this kind of neural network no longer holds any upper bound and is therefore susceptible to adversarial examples. Following similar procedures, we explain why adversarial examples can fool other deep neural networks with skip connections. Furthermore, we demonstrate that a new family of deep neural networks called Neural ODEs (Chen et al., 2018) holds a weaker upper bound. This weaker upper bound prevents the amount of change in the result from being too large. Thus, Neural ODEs have natural robustness against adversarial examples. We evaluate Neural ODEs on two adversarial datasets to show the robustness of Neural ODEs compared with ResNet.

1 INTRODUCTION

Deep neural networks have made great progress in numerous domains of machine learning, especially in computer vision. But Szegedy et al. (2013) found that most of the existing state-of-the-art neural networks are easily fooled by adversarial examples that generated by putting only very small perturbations to the input images. Since realizing the unstability of deep neural networks, researchers have proposed different kinds of methods to defense adversarial examples, such as adversarial training (Ganin et al., 2016), data compression (Dziugaite et al., 2016), and distillation defense (Papernot et al., 2016). But each of these methods is a remedy for the original problem, and none of these methods can solve it fundamentally. For example, Moosavi-Dezfooli et al. (2016) showed that no matter how much adversarial examples are added to training sets, there are new adversarial examples that can successfully attack the adversarial trained deep neural network.

So, avoiding adversarial examples technically cannot solve the most essential problem: why such subtle change in adversarial examples can beat deep neural networks? Meanwhile, it leads to a more important question: how to design deep neural networks that have natural robustness so that they can get rid of malicious adversarial examples.

Early explanations for adversarial examples considered that a smoothness prior is typically valid for kernel methods that imperceptibly tiny perturbations of a given image do not normally change the underlying class, while the smoothness assumption does not hold for deep neural networks due to its high non-linearity (Szegedy et al., 2013). This analysis underlies plain deep neural networks like AlexNet (Krizhevsky et al., 2012). But later than that, Goodfellow et al. (2014) claim adversarial examples are a result of models being too linear rather than too non-linear, they can be explained as a property of high-dimensional dot products. Unfortunately, both of these explanations seem to imply that adversarial examples are inevitable to be avoided by deep neural networks.

On the other hand, we notice that skip connections are widely used in current deep neural networks after the appearance of Highway Network (Srivastava et al., 2015) and ResNet (He et al., 2016). It turns out that the identity mapping in ResNet is formally equivalent to one step of Euler's method which has been used to solve ordinary differential equations. More than that, other kinds of skip connections used by different network architectures can be considered as different numerical methods for solving ordinary differential equations. The link between numerical ordinary differential

equations with deep neural networks can bring us a whole new perspective to explain adversarial examples through the numerical stability analysis.

In this paper, we attempt to utilize the natural property of neural networks to dense adversarial examples. We first analyze how adversarial examples affect the output of neural networks with identity mappings, obtain an upper bound for this kind of neural networks, and find that this upper bound is impractical in actual computations. In the same way, we figure out why adversarial examples can fool commonly used deep neural networks with skip connections. Then, we demonstrate that Neural ODEs hold a weaker upper bound and verify the natural robustness of Neural ODEs on two adversarial datasets. The main contributions of our work are as follows:

- We introduce and formalize the numerical stability analysis for deep neural networks with identity mappings, prove that there is an upper bound for neural networks with identity mappings to constrain the error caused by adversarial noises.
- We provide a new reason why commonly used deep neural networks with skip connections cannot resist adversarial examples.
- We demonstrate that Neural ODEs hold a weaker upper bound which limits the amount of change in the result from being too large. Compare with ResNet, we show the natural robustness of Neural ODEs.

2 Related works

2.1 Adversarial defense

Adversarial training typically uses a robust optimization to generate adversarial examples for training deep neural networks. Madry et al. (2017) take the optimization as a saddle point problem. Zhang et al. (2019) cast adversarial training as a discrete time differential game. Adversarial training can be seen as a data augmentation particularly enhance the robustness to white-box attacks (Tramèr et al., 2017). Zantedeschi et al. (2017) augmented the training sets with examples perturbed using Gaussian noises which can also enhance the robustness to black-box attacks. Besides, Lee et al. (2017) proposed a novel adversarial training method using a generative adversarial network framework.

Modifying the neural networks by using auto encoders, input gradient regularization and distillation can result in robustness against adversarial attacks (Bai et al., 2017; Ross & Doshi-Velez, 2017; Papernot et al., 2016). Besides, there are some biologically inspired deep learning models designed to have natural robustness against adversarial examples. Nayebi & Ganguli (2017) developed a scheme similar to nonlinear dendritic computation to train deep neural networks to make them robust to adversarial attacks. Krotov & Hopfield (2018) proposed Dense Associative Memory (DAM) models and suggested that DAM with higher order energy functions are closer to human visual perception than deep neural networks with ReLUs.

Adversarial defense methods that do not need to augment datasets or modify original neural networks rely on using external models. Akhtar et al. (2018) presented Perturbation Rectifying Network (PRN) as 'pre-input' layers to a targeted model, if a perturbation is detected, the output of the PRN is used for label prediction instead of the actual image.

2.2 The link between ODEs with DNNs

Due to the excellent results achieved by ResNet, lots of hypotheses have merged that try to interpret the role of skip connections in ResNet. Weinan (2017) first observed that deep residual networks can be thought of as a discretization of the continuous dynamical systems. From this point of view, Lu et al. (2018) showed that many existing deep neural networks can be considered as different numerical schemes approximating ordinary differential equations. They showed that PolyNet (Zhang et al., 2017) can be interpreted as backward Euler method, FractalNet (Larsson et al., 2016) is extremely similar to Runge-Kutta methods and RevNet (Gomez et al., 2017) can be conceived as applying Euler's method to a particular dynamic system.

It turns out ODEs can directly apply to deep neural networks. Instead of specifying a discrete sequence of hidden layers, Neural ODEs parameterize the derivative of the hidden state using a neural

network that allows computing output of the network by using a black-box differential equation solver (Chen et al., 2018). Augmented Neural ODEs (Dupont et al., 2019) extended Neural OEDs by allowing the model to use the additional dimensions to learn more complex functions using simpler flows.

3 NUMERICAL STABILITY ANALYSIS FOR DNNS

3.1 RESNET AND EULER'S METHOD

The building block of ResNet is defined as

$$\mathbf{y}_{n+1} = \mathbf{y}_n + f(\mathbf{y}_n; \theta_n) \tag{1}$$

Where $n \in \{0...N(h)-1\}$, $\mathbf{y}_n$ and $\mathbf{y}_{n+1}$ are input and output vectors of the layers considered. The function $f(\mathbf{y}_n; \theta_n)$ represents the residual mapping and θ_n are weights to be learned.

In numerical methods for solving ordinary differential equations, Euler's method is defined by taking this to be exact:

$$\mathbf{y}_{n+1} = \mathbf{y}_n + hf(t_n, \mathbf{y}_n; \theta_n) \tag{2}$$

It can be easily seen that Eqn.(1) is a special case of Euler's method when the step size $h = 1$. The iterative updates of ResNet can be seen as an Euler discretization of a continuous transformation (Lu et al., 2018; Haber & Ruthotto, 2017; Ruthotto & Haber, 2019). When the input data is perturbed by adversarial noises ϵ, we denote the adversarial example by

$$\mathbf{z}_0 = \mathbf{y}_0 + \epsilon \tag{3}$$

To perform stability analysis for ResNet with adversarial example $\mathbf{z}_0$ as input, we define a numerical solution $\mathbf{z}_n$ by

$$\mathbf{z}_{n+1} = \mathbf{z}_n + hf(t_n, \mathbf{z}_n; \theta_n) \tag{4}$$

and provide following theorem to show that the amount of change between $\mathbf{y}_n$ and $\mathbf{z}_n$ holds an upper bound by some assumptions.

Theorem 3.1. *Let $f(t, \mathbf{y}; \theta)$ be a continuous function for $t_0 \leq t \leq b$ and $-\infty < \mathbf{y} < \infty$, and further assume that $f(t, \mathbf{y}; \theta)$ satisfies the Lipschitz condition. Then compare the two numerical solutions $\mathbf{y}_n$ and $\mathbf{z}_n$ as $h \to 0$, there is a constant $c \geq 0$, such that amount of change between $\mathbf{y}_n$ and $\mathbf{z}_n$ satisfies*

$$\max_{0 \leq n \leq N(h)} |\mathbf{z}_n - \mathbf{y}_n| \leq c|\epsilon| \tag{5}$$

Proof. Let $\mathbf{e}_n = \mathbf{z}_n - \mathbf{y}_n, n \geq 0$. Then $\mathbf{e}_0 = \epsilon$, and subtracting Eqn.(2) from Eqn.(4), we obtain

$$\mathbf{e}_{n+1} = \mathbf{e}_n + h[f(t_n, \mathbf{z}_n; \theta_n) - f(t_n, \mathbf{y}_n; \theta_n)] \tag{6}$$

Assume that the derivative function $f(t, \mathbf{y}; \theta)$ satisfied the following Lipschitz condition: there exists $K \geq 0$ such that

$$|f(t, \mathbf{y}_1; \theta) - f(t, \mathbf{y}_2; \theta)| \leq K|\mathbf{y}_1 - \mathbf{y}_2| \tag{7}$$

Taking bounds using Eqn.(7), we obtain

$$|\mathbf{e}_{n+1}| \leq |\mathbf{e}_n| + hK|\mathbf{z}_n - \mathbf{y}_n| \tag{8}$$

$$|\mathbf{e}_{n+1}| \leq (1 + hK)|\mathbf{e}_n| \tag{9}$$

Apply this recursively to obtain

$$|\mathbf{e}_n| \leq (1 + hK)^n |\mathbf{e}_0| \tag{10}$$

Using the inequality $(1 + t)^m \leq e^{mt}$, for any $t \geq -1$, any $m \geq 0$, we obtain

$$(1 + hK)^n \leq \mathbf{e}^{nhK} = \mathbf{e}^{(b-t_0)K} \tag{11}$$

and this implies the main result Eqn.(5). □

Roughly speaking, theorem 3.1 means that a small adversarial perturbation initial value of the problem leads to a small change in the solution, provided that the function $f(t, \mathbf{y}; \theta)$ is continuous and the step size h is sufficiently small.

Obviously, ResNet does not satisfied the assumption of continuity and its step size h always equal to 1. Particularly, the functions $f(t, \mathbf{y}; \theta)$ in ResNet are composite function which contains ReLU activation functions. Although ReLU activation functions $g(x) = max(0, x)$ break the continuity, due to their contractive property, i.e. satisfies $\|g(x) - g(x + \epsilon)\| \leq |\epsilon|$ for all x, ϵ; it follows that

$$\begin{aligned} &\|g_n(x; \theta_n) - g_n(x + \epsilon; \theta_n)\| \\ &= \|max(0, \theta_n x + b_n) - max(0, \theta_n(x + \epsilon) + b_n))\| \\ &\leq \|\theta_n \epsilon\| \leq \|\theta_n\| \|\epsilon\| \end{aligned} \tag{12}$$

This provides a weaker upper bound for ReLU to mitigate the loss of continuity.

Even if the step size h in ResNet can be adjusted by multiply outputs of convolution layers by a chosen constant, but unfortunately, the step size h in ResNet cannot be too small since a very small step size decreases the efficiency in actual computations. We can experimentally show that when step size h is small (such as 10^{-1}, 10^{-2} and 10^{-3}), ResNet has no obvious robustness against adversarial examples, and when step size is very small (such as 10^{-8}, 10^{-9}, 10^{-10}), ResNet is difficult to train, so not only the classification accuracy of adversarial examples but also the accuracy of "clean" examples is quite low.

To summarize, the main reason for ResNet's failure in adversarial examples is the step size $h = 1$ destroys the upper bound given by Eqn.(5) and we cannot find a proper way to deal with it. Thus, when the input of ResNet is perturbed by adversarial noises, the amount of change in the result will become unpredictably large so that ResNet can no longer correctly classify the input.

3.2 Neural ODEs

The residual block can be described as $\mathbf{y}_{n+1} = \mathbf{y}_n + hf(\mathbf{y}_n; \theta_n)$ with step size $h = 1$. When taking smaller steps and add more layers, which in the limit, Neural ODEs parameterize the continuous dynamics of hidden units using an ordinary differential equation specified by a neural network

$$\lim_{h \to 0} \frac{\mathbf{y}_{n+h} - \mathbf{y}_n}{h} = \frac{d\mathbf{y}}{dh} = f(t, \mathbf{y}; \theta) \tag{13}$$

The solution of Eqn.(13) can be computed using modern ODE solvers such as Runge–Kutta (Runge, 1895; Kutta, 1901). Euler's method only uses $f(t_n, \mathbf{y}_n; \theta_n)$, which means from any point on a curve, we can find an approximation of a nearby point on the curve by moving a short distance along a tangent line to the curve. Instead of using a single tangent line, modern numerical ODE methods often use lots of different tangent lines and weighted sum all of them to generate a super tangent line, we denote the super tangent line by $F(t_n, \mathbf{y}_n; \theta_n, f)$.

We can show that modern ODE solvers based on Runge–Kutta are stable under some assumptions, the amount of change between $\mathbf{y}_n$ and $\mathbf{z}_n$ holds an upper bound similar to theorem 3.1.

Theorem 3.2. *Let $f(t, \mathbf{y}; \theta)$ be a continuous function for $t_0 \leq t \leq b$ and $-\infty < \mathbf{y} < \infty$, and further assume that $f(t, \mathbf{y}; \theta)$, $F(t, \mathbf{y}; \theta, f)$ satisfies the Lipschitz condition. Then compare the two numerical solutions $\mathbf{y}_n$ and $\mathbf{z}_n$ as $h \to 0$, there is a constant $\widehat{c} \geq 0$, such that amount of change between $\mathbf{y}_n$ and $\mathbf{z}_n$ satisfies*

$$\max_{0 \leq n \leq N(h)} |\mathbf{z}_n - \mathbf{y}_n| \leq \widehat{c}|\epsilon| \tag{14}$$

Proof. This result can be obtained in analogy with Eqn.(5) in theorem 3.1. □

The neural network in Neural ODEs also use ReLU activation functions. And we already show that Eqn.(12) provides a weaker upper bound for ReLU.

Besides, modern ODE solvers, such as Fehlberg method (Fehlberg, 1969) and DOPRI method (Dormand & Prince, 1980), can adaptively adjust the step size until that the desired tolerance is reached. At each step, two different approximations (for example, a fourth order Runge–Kutta and a fifth

order Runge–Kutta) for the solution are made and compared. If the two answers are in close agreement, the approximation is accepted. If the two answers do not agree to a specified accuracy, the step size is reduced. If the answers agree to more significant digits than required, the step size is increased. So modern ODE solvers provide a weaker upper bound for step size when it does not agree with $h \to 0$.

To summarize, although Neural ODEs cannot hold the strong upper bound provided by Eqn.(14), it still holds a weaker upper bound to constrain the error caused by adversarial noises. We will experimentally show that Neural ODEs is more robust on adversarial examples compared with ResNet.

3.3 Other Deep Neural Networks with Skip Connections

Following similar procedures, we can figure out the reason why other deep neural networks with skip connections cannot resist adversarial examples.

3.3.1 PolyNet

PolyNet (Zhang et al., 2017) introduced a family of models called PolyInception. Each PolyInception module is a polynomial combination of Inception units that can be described as

$$\left(I + F + F^2\right) \cdot \mathbf{y} = \mathbf{y} + F(\mathbf{y}) + F(F(\mathbf{y})) \tag{15}$$

Where $\mathbf{y}$ denotes the input, I the identity operator, and F the nonlinear transform carried out by the residual block, which can also be considered as an operator.

It has been shown that Eqn.(15) can be interpreted as an approximation to one step of the backward (or implicit) Euler method:

$$\mathbf{y}_{n+1} = \mathbf{y}_n + hf(t_{n+1}, \mathbf{y}_{n+1}) \tag{16}$$

And from Eqn.(16) we can get

$$\mathbf{y}_{n+1} = (I - hf)^{-1}\mathbf{y}_n \tag{17}$$

Where $(I - hf)^{-1}$ can be formally rewrite as

$$I + hf + (hf)^2 + \cdots + (hf)^n + \cdots \tag{18}$$

So, there are two differences between the polynomial combination with the backward Euler method:

- The polynomial combination is only a second order approximation to the backward Euler method, the truncation error generated when high order terms are ignored.
- PolyNet faces the same problem with ResNet that the step size $h = 1$. Thus, PolyNet cannot holds the upper bound given by the backward Euler method.

And step size $h = 1$ is the main reason that PolyNet fails to resist adversarial examples.

3.3.2 FractalNet

FractalNet (Larsson et al., 2016) introduced a design strategy for neural network macro-architecture based on self-similarity. The expansion rule generates a fractal architecture with C intertwined columns can be described as

$$f_{C+1}(\mathbf{y}) = \frac{1}{2}(f_C \circ f_C)(\mathbf{y}) + \frac{1}{2}f_1(\mathbf{y}) \tag{19}$$

Where $f_1(\mathbf{y}) = \mathrm{conv}(\mathbf{y})$ and $\circ$ denotes composition.

This expansion rule resembles to the Runge-Kutta method of order 2 (also known as Heun's method)

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \frac{1}{2}h(f(t_n, \mathbf{y}_n) + f(t_{n+1}, \mathbf{y}_n + hf(t_n, \mathbf{y}_n))) \tag{20}$$

So, there are two differences between the expansion rule with Heun's method:

- The expansion rule losses two identity mapping, namely $\mathbf{y}_n$, compared to Heun's method.

- FractalNet faces the same problem with ResNet that the step size $h = 1$. Thus, FractalNet cannot holds the upper bound given by Heun's method.

And step size $h = 1$ is the main reason that FractalNet fails to resist adversarial examples.

3.3.3 REVNET

RevNet (Gomez et al., 2017) introduced a variant of ResNet where each layer's activations can be reconstructed exactly from the next layer's. Each reversible block takes inputs and produces outputs according to the following additive coupling rules

$$\begin{aligned} X_{n+1} &= X_n + f_1\left(Y_n\right) \\ Y_{n+1} &= Y_n + f_2\left(X_{n+1}\right) \end{aligned} \tag{21}$$

This additive coupling rules can be interpreted as Euler's method applies to the following systems of differential equations

$$\begin{aligned} \dot{X} &= f_1(Y, t) \\ \dot{Y} &= f_2(X, t) \end{aligned} \tag{22}$$

RevNet faces the same problem with ResNet that the step size $h = 1$. Thus, RevNet cannot holds the upper bound given by Euler's method for the dynamic systems.

4 EXPERIMENTAL RESULTS

4.1 RESNET WITH STEP SIZE $h \neq 1$

We first evaluate the performance of ResNets with different step sizes on adversarial examples. Adopting the same hyperparameter settings for ResNet56 that presented in He et al. (2016), we train 11 ResNet56s on CIFAR10 datasets with the step size decreases from 10^{-0} to 10^{-10}.

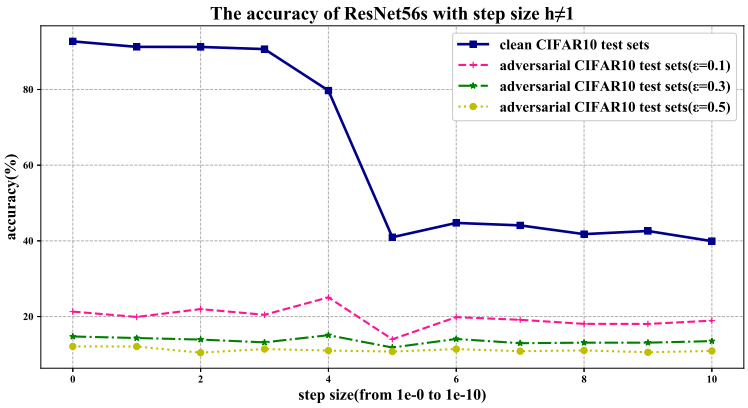

Figure 1: The accuracy of 11 ResNet56s both on clean CIFAR10 test sets and three adversarial CIFAR10 test sets. The adversarial CIFAR10 test sets are generated by Fast Gradient Sign Method with $\epsilon = 0.1, 0.3, 0.5$ on another neural network which has three residual blocks.

Through the numerical stability analysis of ResNet, we know that the classification result should be well behaved when considering small adversarial noises, provided that the step size of h is sufficiently small. In actual computations, however, when we decrease the step size from 10^{-1} to 10^{-10}, ResNet has no obvious robustness against adversarial examples. And when step size is very small, ResNet is difficult to train. Therefore, the accuracy of ResNet56 is considerably low both on clean and adversarial test sets. We consider that ResNet is impossible to have natural robustness against adversarial examples.

4.2 THE NATURAL ROBUSTNESS OF NEURAL ODES

As we mentioned in 3.2, Neural ODEs define a continuous dynamic system, which is equivalent to ResNet with step size $h \to 0$. And Neural ODEs hold a weaker upper bound to constrain the error caused by adversarial noises. In this section, we evaluate the performance of Neural ODEs on two adversarial datasets generated by Fast Gradient Sign Method (FGSM) and Project Gradient Descent (PGD). Madry et al. (2017) pointed out that PGD is the strongest attack utilizing the local first order information about the network.

Table 1 summarizes our experiment settings for training Neural ODEs. We only change some hyperparameters in Neural ODEs, without taking any help of adversarial defense methods.

Training Configurations	MNIST	CIFAR10
training method	Adam	Adam
ODEslover	dopri5	dopri5
convolution layers	4	5
convolution channels	128	256
start time t_0	0	0
stop time t_1	100	100
training epochs	100	200
initial learning rate*	10^{-3}	10^{-4}
mini-batches size	32	32

Table 1: The training configurations of Neural ODEs on MNIST and CIFAR10. *Every 50 epochs, the learning rate is reduced to half.

Table 2 shows that Neural ODEs have natural robustness against adversarial examples to some extent. The accuracy of adversarial MNIST datasets generated by FGSM only drops by less than 2% for all ϵ. Even the strongest first order attack method PGD can hardly beat Neural ODEs on MNIST. Although the performance of Neural ODEs on clean CIFAR10 test sets is worse than state-of-the-art networks like ResNet, it still works well on adversarial CIFAR10 test sets generated by FGSM. However, Neural ODEs also fail to resist adversarial examples generated by PGD with large ϵ. We consider that this is because the weaker upper bound is difficult to handle this situation. Overall, Neural ODEs are more stable when facing adversarial noises compared with ResNet.

Model	Datasets		clean	$\epsilon = 0.1$	$\epsilon = 0.2$	$\epsilon = 0.3$	$\epsilon = 0.4$	$\epsilon = 0.5$
ResNet50	MNIST	clean	98.94					
		FGSM		10.37	9.71	9.74	9.74	9.74
		PGD		0.05	0	0	0	0
Neural ODEs	MNIST	clean	99.40					
		FGSM		97.52	97.50	97.46	97.52	97.48
		PGD		95.13	93.62	90.55	84.64	75.23
ResNet50	CIFAR10	clean	93.54					
		FGSM		12.79	10.28	8.57	8.35	8.75
		PGD		0.03	0	0	0	0
Neural ODEs	CIFAR10	clean	71.64					
		FGSM		50.28	50.33	50.42	50.46	50.46
		PGD		35.26	19.03	12.77	10.48	9.12

Table 2: The performance of Neural ODEs and ResNet50 on two adversarial datasets. It is worth to mention that there are no common training configurations for Neural ODEs on CIFAR10, Dupont et al. (2019) had reported that the accuracy of Neural ODEs on CIFAR10 test sets is $53.7\% \pm 0.2$, our experiment result is better than it.

5 CONCLUSION AND DISSCUSSION

In this paper, we highlighted and analyzed the natural robustness of Neural ODEs. We proved there are two similar upper bounds for ResNet and Neural ODEs under assumptions of continuity and

step size $h \to 0$. We showed that it is the step size $h = 1$ causes ResNet and other three deep neural networks with skip connections fail on adversarial examples. Neural ODEs define a continuous dynamic system with step size $h \to 0$. However, in actual computations, Modern ODE solvers adaptively adjust the step size to solve the continuous dynamic system and this brings us a weaker upper bound for Neural ODEs. We experimentally showed that Neural ODEs is more robust on adversarial examples compared with ResNet.

Due to the similarity between DNNs and numerical methods for solving ODEs, we believe that we can learn from the numerical stability analysis to design more robust deep neural networks.

REFERENCES

Naveed Akhtar, Jian Liu, and Ajmal Mian. Defense against universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3389–3398, 2018.

Wenjun Bai, Changqin Quan, and Zhiwei Luo. Alleviating adversarial attacks via convolutional autoencoder. In *2017 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, pp. 53–58. IEEE, 2017.

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

John R Dormand and Peter J Prince. A family of embedded runge-kutta formulae. *Journal of computational and applied mathematics*, 6(1):19–26, 1980.

Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. Augmented neural odes. In *Advances in Neural Information Processing Systems*, pp. 3140–3150, 2019.

Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M Roy. A study of the effect of jpg compression on adversarial images. *arXiv preprint arXiv:1608.00853*, 2016.

Erwin Fehlberg. *Low-order classical Runge-Kutta formulas with stepsize control and their application to some heat transfer problems*, volume 315. National aeronautics and space administration, 1969.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

Aidan N Gomez, Mengye Ren, Raquel Urtasun, and Roger B Grosse. The reversible residual network: Backpropagation without storing activations. In *Advances in neural information processing systems*, pp. 2214–2224, 2017.

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

Eldad Haber and Lars Ruthotto. Stable architectures for deep neural networks. *Inverse Problems*, 34(1):014004, 2017.

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*, pp. 770–778, 2016.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Dmitry Krotov and John Hopfield. Dense associative memory is robust to adversarial inputs. *Neural computation*, 30(12):3151–3167, 2018.

Wilhelm Kutta. Beitrag zur naherungsweisen integration totaler differentialgleichungen. *Z. Math. Phys.*, 46:435–453, 1901.

Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. *arXiv preprint arXiv:1605.07648*, 2016.

Hyeungill Lee, Sungyeob Han, and Jungwoo Lee. Generative adversarial trainer: Defense to adversarial perturbations with gan. *arXiv preprint arXiv:1705.03387*, 2017.

Yiping Lu, Aoxiao Zhong, Quanzheng Li, and Bin Dong. Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations. In *International Conference on Machine Learning*, pp. 3276–3285. PMLR, 2018.

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.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2574–2582, 2016.

Aran Nayebi and Surya Ganguli. Biologically inspired protection of deep networks from adversarial attacks. *arXiv preprint arXiv:1703.09202*, 2017.

Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 582–597. IEEE, 2016.

Andrew Slavin Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. *arXiv preprint arXiv:1711.09404*, 2017.

Carl Runge. Über die numerische auflösung von differentialgleichungen. *Mathematische Annalen*, 46(2):167–178, 1895.

Lars Ruthotto and Eldad Haber. Deep neural networks motivated by partial differential equations. *Journal of Mathematical Imaging and Vision*, pp. 1–13, 2019.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

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.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

E Weinan. A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 5(1):1–11, 2017.

Valentina Zantedeschi, Maria-Irina Nicolae, and Ambrish Rawat. Efficient defenses against adversarial attacks. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 39–49, 2017.

Dinghuai Zhang, Tianyuan Zhang, Yiping Lu, Zhanxing Zhu, and Bin Dong. You only propagate once: Accelerating adversarial training via maximal principle. In *Advances in Neural Information Processing Systems*, pp. 227–238, 2019.

Xingcheng Zhang, Zhizhong Li, Chen Change Loy, and Dahua Lin. Polynet: A pursuit of structural diversity in very deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 718–726, 2017.