Can Neural ODEs Offer Free Robustness ?

Dario Shariatian
University of Oxford

Abstract

Recent studies have demonstrated that Neural ODEs are intrinsically more robust against adversarial attacks compared to vanilla DNNs, at comparable accuracy. Inspired by dynamical system theory, we look at theoretical explanation trying to support this case, and devise effective methods accordingly. Indeed, as the picture gets more complex, initial justifications for robustness reveal unsatisfactory and additional expressivity concerns arise. Finally, we look at the effectiveness of stochastic noise regularization in regards to the last points.

1 Introduction

Recent research has bridged dynamical systems with neural networks as pioneered in [1]. Initial work [16] suggest that Neural ODEs offer natural robustness against adversarial attacks at equal accuracy. Many contradicting studies ([8], [16], [7]) expose theoretical limitations while readapting the argument with Lyapunov type theorems. We'll explore the SODEF [8] perspective. Moreover, expressivity issues have been raised building on the very same arguments previously explaining model strength.

Contribution We try to see if stochastic regularization can improve robustness and expressivity. Effectiveness against adversarial robustness must be carefully questioned in light of [7]. Indeed most architecture are tested with white-box attacks that may fail because of gradient masking caused by the ODE solvers (adptive steps). We will attack vanilla/stochastic SODEF [8] accordingly to see if the empirical claim of [7] holds. To our knowledge, this work is novel.

2 What are Neural ODEs ?

Neural ODEs [1] were inspired by the ResNet forward equation between hidden states:

$$z_{t+1} = z_t + f_{\theta,t}(z_t) \tag{1}$$

which can be seen as Euler discretizations of a continuous transformation. Rewriting the equation in bounded time $[0,1]$ gives $z_{t+1/n} = h_t + \frac{1}{n} f_{\theta,t}(z_t)$ for which we obtain a natural continuous model when taking $n \to +\infty$:

$$\frac{dz_t}{dt} = f_{\theta,t}(z_t) \tag{2}$$

It is a parametrization of an ODE by a neural network. Thus loss L can be written

$$L(z(1)) = L\left(z(0) + \int_0^1 f_{\theta,t}(z(t))dt\right) = L\left(\text{ODESolve}(z(0), f, \theta)\right)$$

Efficient solvers and optimization methods are readily available. In particular, neural ODEs rather use the **adjoint method** instead of backpropagation for model weight updates. We recommend reading Appendix B of [1]. This **adjoint method** scales linearly with problem size, has low memory cost ($O(1)$), and explicitly controls numerical error. Let us outline two immediate benefits.

(Memory efficiency) Not storing any intermediate quantities on the forward/backward pass allows us to train models with constant memory cost as a function of depth, a major bottleneck of training deep models. **(Adaptive computation)** Modern ODE solvers adapt their evaluation strategy to achieve a desired level of accuracy, using a good proxy such as the number of function evaluations. As they increase, the evaluations become closer in time, emulating a deeper residual network. This allows the ODE solver to adaptively determine the depth needed for the model's expressivity. After training, the accuracy can be reduced for real-time or low-power applications.

3 Precise Integration Introduce Intrinsic Robustness ?

Empirical studies suggest that at comparable accuracy, neural ODEs show improved robustness. [16] found that ODE networks with natural training are more robust against adversarial examples and gaussian noise compared to conventional neural networks (while still being weaker than state-of-the-art models obtained by adversarial training, like TRADES [9]). The initial intuition on these results come from the following property of ODEs:

Theorem 1 *(non-crossing of integral curves) Let z_1, z_2 be two solutions of ODE* (2) *with different initial conditions $z_1(0) \neq z_2(0)$. If $f_{\theta,t}$ is continuous in time t and globally Lipschitz in z then $z_1(t) \neq z_2(t)$ for all $t \geq 0$.*

Consider a 1-D system and take z_1, z_2, with $z_1(0) < z_2(0)$. Then $\forall z$ s.t $z_1(0) < z(0) < z_2(0)$, we must have $z_1(1) < z(1) < z_2(1)$. Thus all the possible perturbations in the initial condition of our problem constrained in some ball can be uniformly bounded. However, a lemma can be deceiving:

Lemma 1 *(Gronwall's Lemma) Let $U \subset \mathbb{R}^d$ be an open set, $f : U \times [0,T] \mapsto \mathbb{R}^d$ and $z_1, z_2 : [0,T] \mapsto U$ satisfy the ODE* (2) *parametrized by f. If $f(\cdot, t)$ is C Lipschitz for all t then*

$$\|z_1(t) - z_2(t)\| \leq \|z_1(0) - z_2(0)\| e^{Ct} \tag{3}$$

And the bound can be reached (take $f(x,t) = Cx$. Then $z(t) = z(0)e^{Ct}$). This is an exponential expansion from initial conditions, at a rate approximately the regularity of f. This certainly hinders robustness. Approaches have been proposed to address the issue. [3] make use of optimal transport and physics inspired measures to reduce instabilities. The method we'll use takes another approach.

4 A More Complex Picture with Lyapunov Stability: SODEF

[8] takes another perspective. Lyapunov stability theorems let us build a framework where small perturbations around inputs of interests actually converge to the same output with exponential contraction of distance $O(e^{-Ct})$. In this framework, we will focus on a classification tasks between L classes, and on the time-invariant case : $f_{\theta,t} = f_\theta$. First, the Hartman-Grobman theorem [5] introduces a nice setting to rather study the linearization of the dynamics:

Theorem 2 *(Hartman-Grobman). Consider a dynamical system as described in* (2) *for some $f \in \mathcal{C}^1(\mathbb{R}^n, \mathbb{R}^n)$. Suppose the map has a hyperbolic equilibrium state z^*, i.e $f(z^*) = 0$, and that $\Delta f(z^*)$ has no eigenvalue with zero real part. Then there exists a neighbourhood U_{z^*}, a homeomorphism $g : U_{z^*} \mapsto \mathbb{R}^n$ s.t $g(z^*) = 0$ and in U_{z^*} the flow of $\frac{dz}{dt}(t) = f(z(t))$ is topologically conjugate by the map $\hat{z} = g \circ z$ to the flow of its linearization $\frac{d\hat{z}}{dt}(t) = \Delta f(z^*)(\hat{z}(t))$.*

Now we define stability in the Lyapunov sense in linear time-invariant systems:

Theorem 3 *(Lyapunov Stability) Define an ODE with constant matrix A by $\frac{d\hat{z}}{dt}(t) = A(\hat{z}(t))$. Then we have the following equivalences: 1) (Stability) Every finite initial state $\hat{z}(0)$ excites a bounded response iff all the eigenvalues of A have negative real part and those with zero real part are simple roots of the minimal polynomial of A. 2) (Asymptotic stability) These states approach 0 as $t \to +\infty$ iff the eigenvalues of A are strictly negative.*

Thus Lyapunov stable equilibrium points are naturally robust to perturbations and act as noise filters. In view of implementation we further introduce the following theorem:

Theorem 4 *(Strictly Diagonally Dominant) Let $A \in \mathcal{M}_{n,n}(\mathbb{C})$ with negative diagonal elements. Suppose that A is strictly diagonally dominant, i.e $\forall i, |A_{ii}| > \sum_{j \neq i} |A_{ij}|$. Then A is non singular and all its eigenvalues have strictly negative real parts (direct application of Gershgorin disks).*

Lyapunov-stable equilibrium points for different classes may however locate near each other. Each stable neighborhood may be very small, leading to poor adversarial defense. Thus we would like to map our L stable zones around our L classes far of each other. We thus introduce a final FC (Fully Connected) layer represented by a matrix V; its purpose is to map $z(0)$ to v_l if $z(0)$ belongs to class l, where $V = (v_1, ..., v_l)$. We just have to assure that these vectors are sufficiently far from each other, either by selecting $V \in \mathcal{O}(n)$ orthogonal, or by minimizing $a(V) = \max_{i \neq j} v_i^T v_j$ (see [8], corollary 1). Here the input $x \in \mathcal{X}$ is passed through a feature extractor h_ϕ to obtain the embedding $z(0)$. A

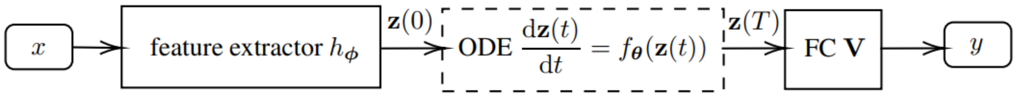

x
feature extractor h_ϕ
$\mathbf{z}(0)$
ODE $\frac{d\mathbf{z}(t)}{dt} = f_\theta(\mathbf{z}(t))$
$\mathbf{z}(T)$
FC $\mathbf{V}$
y

Figure 1: SODEF (Stable neural ODE for deFense) Architecture [8]

neural ODE defined by f_θ stabilizes that representation, obtaining $z(T)$ (usually $T = 1$). This is passed through the FC layer, parametrized by V, to generate a prediction vector y. We introduce the following objective function:

$$\begin{aligned} &\min_{\theta,\phi,} \ \mathbb{E}_\mu l(V^T z(T), y) \\ &\text{s.t } (1) \ \mathbb{E}_\mu \|f_\theta(z(0))\| < \epsilon, \ f_\theta \in \mathcal{C}^1 \\ &\quad\ (2) \ \mathbb{E}_\mu \|\Delta f_\theta(z(0))_{ii}\| < 0 \\ &\quad\ (3) \ \mathbb{E}_\mu [|\Delta f_\theta(z(0))_{ii}| > \sum_{j \neq i} |\Delta f_\theta(z(0))_{ij}|] > 0, \ \forall i \end{aligned} \tag{4}$$

which translates exponential stability. Let's analyze the behaviour of this approach.

Assumption 1. The embedded features $z(0)$ admit conditional distribution μ_l for each class l. They each admit a compact support denoted by E_l.

Assumption 2. The support of each class is pairwise disjoint: $E_i \cap E_j = \emptyset$ if $i \neq j$.

These are reasonable in the context of images for instance. Finally we introduce an existence theorems that justifies our construction. The next lemma is the backbone of our main result.

Lemma 2 *Given k distinct points z_i and matrices A_i, there exists a function $f \in \mathcal{C}^1$ s.t $f(z_i) = 0$ and $\Delta f_\theta(z_i) = A_i$*

Theorem 5 *Suppose the last assumptions. If μ is not a continuous uniform measure on E_l for each l, then 1) the function space satisfying the constraints 1 to 3 is non empty for all $\varepsilon > 0$. 2) If the restriction of μ to any open set $O \subset E_l$ is not a continuous uniform measure, then there exists functions in this space s.t each E_l contains at least one Lyapunov-stable point.*

Moreover, f_θ need only approximate a $\mathcal{C}^1$ function, which is in reach of deep learning architectures, even for shallow networks [6]. The following papers offer similar approaches relying on the same properties. The authors in [14] define only another objective function, with no constraints, that aim at improving stability by implicitly bounding the eigenvalues largest real part, modulating a factor κ. In [7], the approach is even more similar, as the authors introduce a 2-layer net for f_θ that automatically produces Lyapunov-stable dynamics; they implicitly use the Hartman-Grobman theorem.

5 Issues in Expressivity

Authors in [2] point out another problem with Neural ODEs, that once more show precise integration in a bad light; the non crossing integral curves property is a flaw in expressivity. Write $\psi_\theta : z(0) \mapsto z(T)$ where z evolves according to the dynamics described by (2) with f_θ. Then ψ_θ is a homeomorphism (standard result in differential calculus). Thus the mapping preserves the topology of the input space. In particular it cannot tear a connected region apart; this is exemplified in the following result. Let $0 < r_1 < r_2 < r_3$, $g_d \in \mathcal{C}(\mathbb{R}^d, \mathbb{R})$, $d \geq 1$, s.t:

$$\begin{cases} g(x) = -1 & \text{if } \|x\| \leq r_1 \\ g(x) = 1 & \text{if } r_2 \leq \|x\| \leq r_3 \end{cases}$$

Proposition Neural ODEs cannot represent g_d.

This is a very insightful result. Comparing with ResNets, authors explain the discretization error introduced by the Euler discretization makes it possible to learn g_1, for instance.

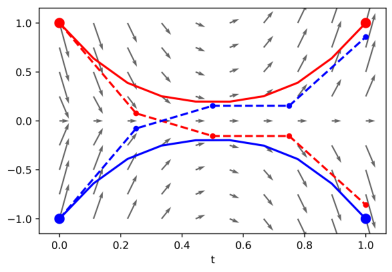


Figure 2: [2] Path of a Neural ODE (solid lines) vs ResNet approximation (dashed lines) for g_1

This is still evidence for limitations of the Resnet architecture too, as the occasional path crossing is unlikely and fortunate. To alleviate this problem, the original authors introduce a simple solution that consists in lifting points to additional dimensions, thus instead solving

$$\frac{d}{dt}\begin{pmatrix} z(t) \\ a(t) \end{pmatrix} = f_{\theta,t}\begin{pmatrix} z(t) \\ a(t) \end{pmatrix}, \quad \begin{pmatrix} z(0) \\ a(0) \end{pmatrix} = \begin{pmatrix} z(0) \\ 0 \end{pmatrix} \tag{5}$$

This solution works well for the use cases presented by [2], and shows better accuracy and generalization at lower computational cost, presumably because the learned flow is simpler and smoother.

6 Stochastic Noise for Expressivity and Stability: From ODE to SDE

Rather than augmentation, we propose an easy drop-in approach to implement. The mainstream deep learning literature [9] frequently uses stochasticity as a regularizer. A neural SDE may likewise be treated as a regularised neural ODE, as is introduced in [10]. Instead of (2), describe the dynamics by

$$dz_t = f_{\theta,t}(z_t)dt + G_{\omega,t}(z_t)dW_t \tag{6}$$

where W_t is a standard Wiener process (see [13] for a refresher on stochastic calculus). This is very simple to implement as SDESolver for neural SDEs are readily available. As long as f and G satisfy some very generic regularity conditions (Lipschitz-ity), which we can assume they will, a unique stable solution exists. Moreover, there exists a reformulation of Lyapunov stability in this setting:

Definition 1 *(Lyapunov stability of SDE) The solution of z as defined in* (6) *is a.s exponentially stable if* $\limsup_{t\to\infty} \frac{1}{t}\log\|z_t\| < 0$ *a.s for all* $z(0)$.

Theorem 6 *(Case $G_{\omega,t}(z_t) = \sigma z_t$) [10] if $f_{\theta,t}$ is L-Lipschitz then* (6) *has a unique solution satisfying* $\limsup_{t\to\infty} \frac{1}{t}\log\|z_t\| \le -(\frac{\sigma^2}{2} - L)$ *a.s for any* $z(0)$.

It can be nice to think about the geometric Brownian motion, which corresponds to the latter case, while further assuming $f_{\theta,t}(z_t) = \lambda z_t$. Then $z_t = \exp((\lambda - \sigma^2/2)t + W_t)$, and the result is intuitive. Thus, increasing the diffusion coefficient brings balance between exponential stability and effectiveness of the model, as it increases information loss. We additionally conjecture that Brownian noise has the ability to violate the non crossing property and thus further increase expressivity.

Replicating Dropout [10] determines that the best model for our diffusion G is a replication of dropout ([15]). Let's look at ResNet using dropout. Nodes are dropped iid with probability $1-p$:

$$z_{t+1} = z_t + f_{\theta_n}(z_n) \odot \frac{\gamma_n}{p} = z_t + f_{\theta_n}(z_n) + f_{\theta_n}(z_n) \odot (\frac{\gamma_n}{p} - I) \tag{7}$$

where $\gamma_n \overset{iid}{\sim} \mathcal{B}(p)$. We divide by p to obtain a unit expectation. Write $B = (\frac{\gamma_n}{p} - I)$. See how $\mathbb{E}(B) = 0$, $\text{Var}(B) = \frac{1-p}{p}$. Thus equating the first two moments we obtain a good approximation by setting $B \sim \sqrt{\frac{1-p}{p}}\mathcal{N}(0,1)$. Combining with (7), the SDE with dropout becomes:

$$dz_t = f_{\theta,t}(z_t)dt + \sqrt{\frac{1-p}{p}} f_{\theta,t}(z_t)dW_t \tag{8}$$

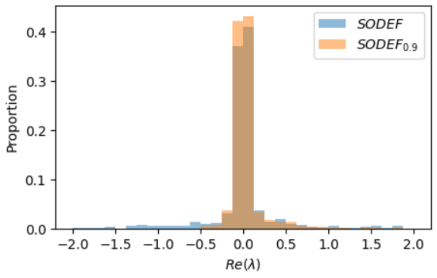
Interestingly, the apparition of an SDE in the Resnet equation need not require dropout or even a probabilistic variable; the insightful work of [12] shows that Resnets initialized with iid parameters have a non-trivial large depth regime only when following SDE dynamics.

7 Experiment: Does Stochastic Noise Improve SODEF Stable Solutions ?

In light of our study, we want to see if introducing stochastic noise can improve accuracy or robustness. We use the same setup as the experiment on MNIST in [8] (implementing stochasticity using `torchsde`): network = 8 layers pre-trained ResNet, 512x64 FC layer, trainable 64x64 SODEF, 64x10 FC layer with orthogonal matrix V; optimizer=Adam with batch size 128 and learning rate 0.001; adjoint method for the ODE and adjoint_reversible_heun method for the SDE with **fixed step size** and integration time $T = 10$, adversarial = $\mathcal{L}^2$ PGD ([11]) and gradient free SGSA ([4]) with $\varepsilon = 0.5$. We train only SODEF, the whole network, and only SODEF again. We denote SODEF by $SODEF_p$, $p = 1, 0.99, 0.9, 0.75$ being the equivalent dropout rate as defined in (8) ($p = 1$ being the usual SODEF). We train each $SODEF_p$ initializing at f_θ corresponding to $SODEF_1$ on a total integration time of $T = 10$, for increased stability. After training, the eigenvalues of the gradient of f_θ for SODEF and $SODEF_{0.9}$ are examined on the same 100 training points chosen uniformly at random.

p	Clean	PGD	SPSA
1	96.2	45.2	40.5
0.99	97.0	46.0	40.1
0.9	**98.5**	**51.2**	**49.4**
0.75	89.1	30.0	35.8

Model performance (accuracy)



Δf_θ eigenvalues real part distribution on training points

Stochastic noise can improve the expressivity of the network, as demonstrated by the increased accuracy ($p = 0.9$). The distribution of real parts of Δf_θ eigenvalues suggests that stochasticity has smoothed the function and reduced high expansion factors ($Re(\lambda) > 0$), consistent with the Lyapunov stability theorem. However, this comes at the expense of stability factors ($Re(\lambda) < 0$), as stable zones are more easily escaped by the Wiener process. This supports the idea of integral flow crossing, which in turn supports expressivity. Furthermore, the model demonstrates robustness to adversarial attacks, even for black-box methods (even though integration time has to be increased to $T \geq 5$), suggesting that robustness is not a fluke, as was suggested in previous research (e.g., [7]).

8 Conclusion

Empirical results and selected theorems suggest that the structure of ODEs may offer robustness guarantees, and that the concurrent expressivity loss can be effectively addressed through the use of stochastic regularization. Recent studies such as [12] have also recognized the natural occurrence of SDEs in conventional deep learning settings, further justifying their appeal. However, it can be challenging to obtain significant results and effective methods using ODEs, as they are typically slower [9] and less effective than other techniques such as TRADES. Additionally, the current theoretical framework for ODEs is lacking, such as the limitations of Lyapunov-type theorems and the breakdown of existence theorems when considering absolutely continuous measures [8], which is more than undesirable (one could think of the angle of a face in image data). Further empirical studies on other datasets and data distributions, as well as the exploration of stochastic regularization, are necessary to fully understand the potential of neural ODEs.

References

[1] Tian Qi Chen et al. "Neural Ordinary Differential Equations". In: *CoRR* abs/1806.07366 (2018). arXiv: 1806.07366. URL: http://arxiv.org/abs/1806.07366.

[2] Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. *Augmented Neural ODEs*. 2019. DOI: 10.48550/ARXIV.1904.01681. URL: https://arxiv.org/abs/1904.01681.

[3] Chris Finlay et al. *How to train your neural ODE: the world of Jacobian and kinetic regularization*. 2020. DOI: 10.48550/ARXIV.2002.02798. URL: https://arxiv.org/abs/2002.02798.

[4] Chuan Guo et al. "Simple Black-box Adversarial Attacks". In: *CoRR* abs/1905.07121 (2019). arXiv: 1905.07121. URL: http://arxiv.org/abs/1905.07121.

[5] Philip Hartman. "A lemma in the theory of structural stability of differential equations". In: 1960.

[6] Kurt Hornik. "Approximation capabilities of multilayer feedforward networks". In: *Neural Networks* 4 (1991), pp. 251–257.

[7] Yifei Huang et al. "Adversarial Robustness of Stabilized Neural ODE Might be from Obfuscated Gradients". In: *Proceedings of the 2nd Mathematical and Scientific Machine Learning Conference*. Ed. by Joan Bruna, Jan Hesthaven, and Lenka Zdeborova. Vol. 145. Proceedings of Machine Learning Research. PMLR, 16–19 Aug 2022, pp. 497–515. URL: https://proceedings.mlr.press/v145/huang22a.html.

[8] Qiyu Kang et al. "Stable Neural ODE with Lyapunov-Stable Equilibrium Points for Defending Against Adversarial Attacks". In: *CoRR* abs/2110.12976 (2021). arXiv: 2110.12976. URL: https://arxiv.org/abs/2110.12976.

[9] Patrick Kidger. "On Neural Differential Equations". In: *CoRR* abs/2202.02435 (2022). arXiv: 2202.02435. URL: https://arxiv.org/abs/2202.02435.

[10] Xuanqing Liu et al. "Neural SDE: Stabilizing Neural ODE Networks with Stochastic Noise". In: *CoRR* abs/1906.02355 (2019). arXiv: 1906.02355. URL: http://arxiv.org/abs/1906.02355.

[11] Aleksander Madry et al. *Towards Deep Learning Models Resistant to Adversarial Attacks*. 2017. DOI: 10.48550/ARXIV.1706.06083. URL: https://arxiv.org/abs/1706.06083.

[12] Pierre Marion et al. *Scaling ResNets in the Large-depth Regime*. 2022. DOI: 10.48550/ARXIV.2206.06929. URL: https://arxiv.org/abs/2206.06929.

[13] Peter Tankov Nizar Touzi. "Calcul Stochastique en Finance". In: *CMAP, Ecole Polytechnique* (2004).

[14] Ivan Dario Jimenez Rodriguez, Aaron D. Ames, and Yisong Yue. "LyaNet: A Lyapunov Framework for Training Neural ODEs". In: *CoRR* abs/2202.02526 (2022). arXiv: 2202.02526. URL: https://arxiv.org/abs/2202.02526.

[15] Nitish Srivastava et al. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

[16] Hanshu Yan et al. "On Robustness of Neural Ordinary Differential Equations". In: *CoRR* abs/1910.05513 (2019). arXiv: 1910.05513. URL: http://arxiv.org/abs/1910.05513.