ResNets Ensemble via the Feynman-Kac Formalism for Adversarial Defense

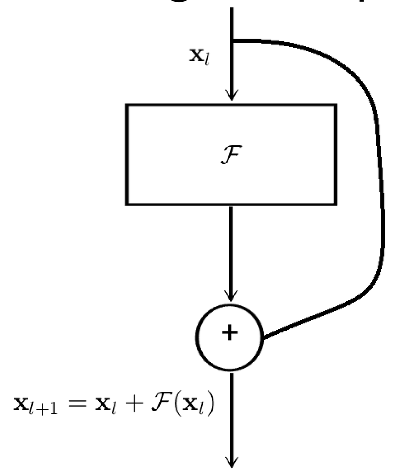
Bao Wang, Binjie Yuan, Zuoqiang Shi, Stanley J. Osher

Transport Equation Modeling of ResNet

Residual mapping:

$$\mathbf{x}_{l+1} = \mathcal{F}(\mathbf{x}_l, \mathbf{w}_l) + \mathbf{x}_l, \ l = 0, 1, \cdots, L-1,$$

with $\mathbf{x}_0 = \hat{\mathbf{x}} \in T \subset \mathbb{R}^d$ being a data point, $\mathbf{w}_l$ is the parameters to learn.



Continuum limit:

$$\frac{d\mathbf{x}(t)}{dt} = \overline{F}(\mathbf{x}(t), \mathbf{w}(t)), \quad \mathbf{x}(0) = \hat{\mathbf{x}}.$$

The above ODE models the data flow of **each** data, and it can be viewed as the characteristics of the following transport equation, which can be used to describe the evolution of the **whole** data distribution via ResNet.

Transport equation model of ResNet:

Forward propagation: compute $u(\hat{\mathbf{x}}, 0)$ along the characteristics of:

$$\begin{cases} \frac{\partial u}{\partial t}(\mathbf{x}, t) + F(\mathbf{x}, \mathbf{w}(t)) \cdot \nabla u(\mathbf{x}, t) = 0, & \mathbf{x} \in \mathbb{R}^d, \\ u(\mathbf{x}, 1) = f(\mathbf{x}), & \mathbf{x} \in \mathbb{R}^d, \end{cases}$$

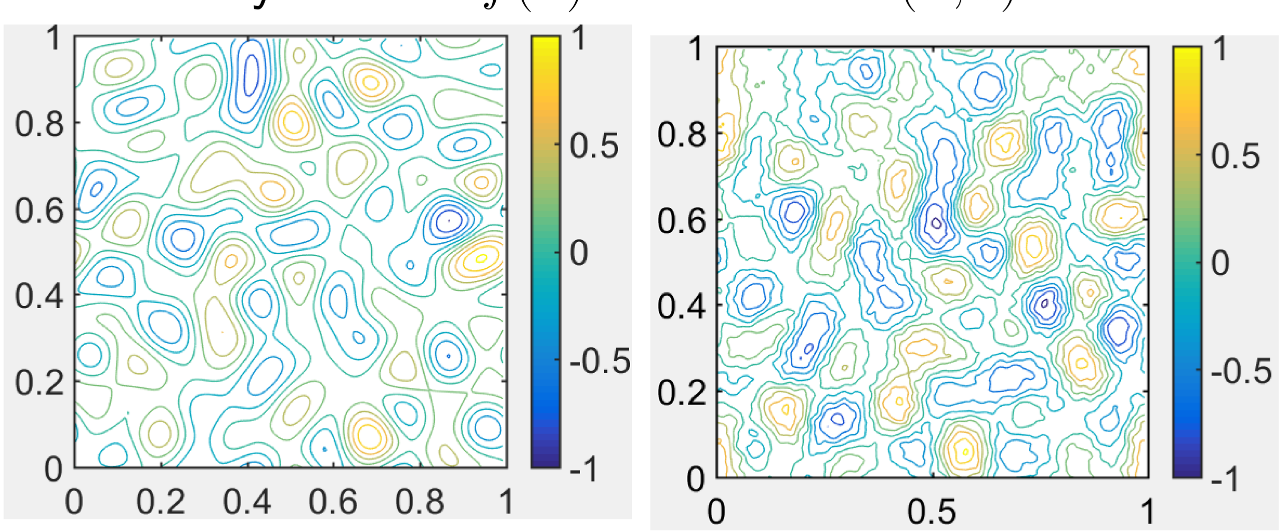
where $f(\mathbf{x})$ is the output activation.

Backward propagation: find the optimal control, $F(\mathbf{x}, \mathbf{w}(t))$, for

$$\begin{cases} \frac{\partial u}{\partial t}(\mathbf{x}, t) + F(\mathbf{x}, \mathbf{w}(t)) \cdot \nabla u(\mathbf{x}, t) = 0, & \mathbf{x} \in \mathbb{R}^d, \\ u(\mathbf{x}, 1) = f(\mathbf{x}), & \mathbf{x} \in \mathbb{R}^d \\ u(\mathbf{x}_i, 0) = y_i, & \mathbf{x}_i \in T. \end{cases}$$

Adversarial Vulnerability – Interpretation

Irregularity of $u(\mathbf{x}, 0)$ which is used to classify the data!



Improve Robustness via Diffusion

We add a diffusion term to the transport equation model above, resulting in the following convection-diffusion equation (CDE)

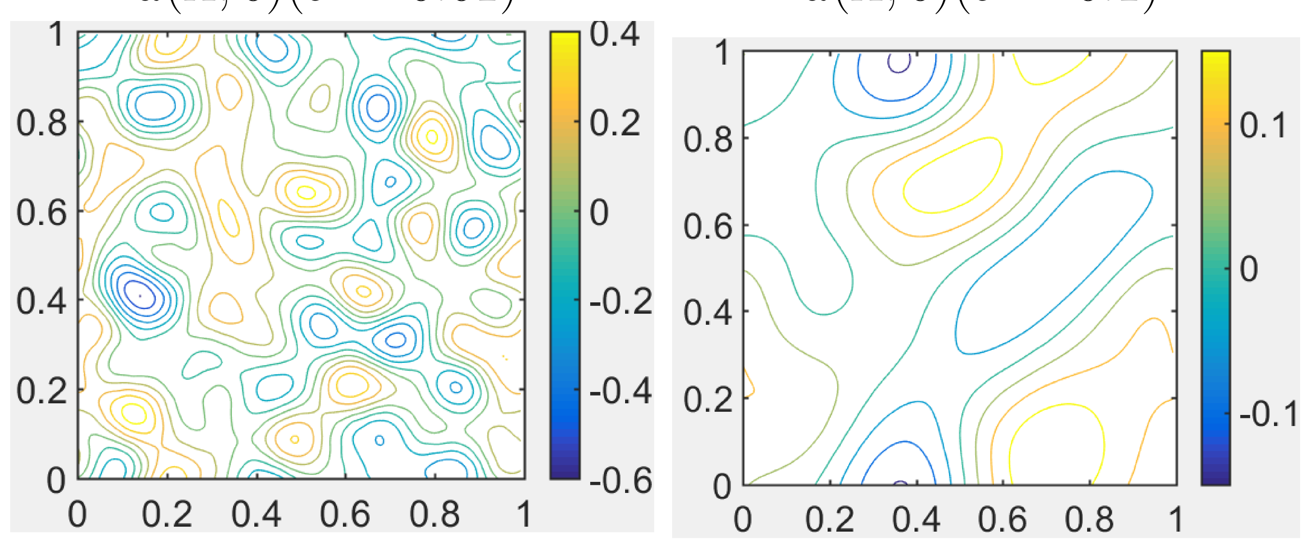
$$\begin{cases} \frac{\partial u}{\partial t}(\mathbf{x}, t) + \overline{F}(\mathbf{x}, \mathbf{w}(t)) \cdot \nabla u(\mathbf{x}, t) + \frac{1}{2}\sigma^2 \Delta u(\mathbf{x}, t) = 0, \ \mathbf{x} \in \mathbb{R}^d, \ t \in [0, 1), \\ u(\mathbf{x}, 1) = f(\mathbf{x}). \end{cases}$$

Stability Theorem Let $\overline{F}(\mathbf{x}, t)$ be Lipschitz in both $\mathbf{x}$ and t, and $f(\mathbf{x})$ be a bounded function. Then, for any small perturbation δ, we have

$$|u(\mathbf{x} + \delta, 0) - u(\mathbf{x}, 0)| \leq C \left(\frac{\|\delta\|_2}{\sigma}\right)^\alpha,$$

for some constant $\alpha > 0$ if $\sigma \leq 1$. Here, $\|\delta\|_2$ is the ℓ_2 norm of δ, and C is a constant that depends on d, $\|f\|_\infty$, and $\|\overline{F}\|_{L^\infty_{\mathbf{x},t}}$.

Diffusion can smooth the decision boundary!



Feynman-Kac Formalism Principled Robust ResNets Ensemble

We can represent $u(\mathbf{x}, 0)$ of the CDE by the Feynman-Kac formula

$$u(\hat{\mathbf{x}}, 0) = \mathbb{E}\left[f(\mathbf{x}(1)) | \mathbf{x}(0) = \hat{\mathbf{x}}\right],$$

where $\mathbf{x}(t)$ is an Itô process,

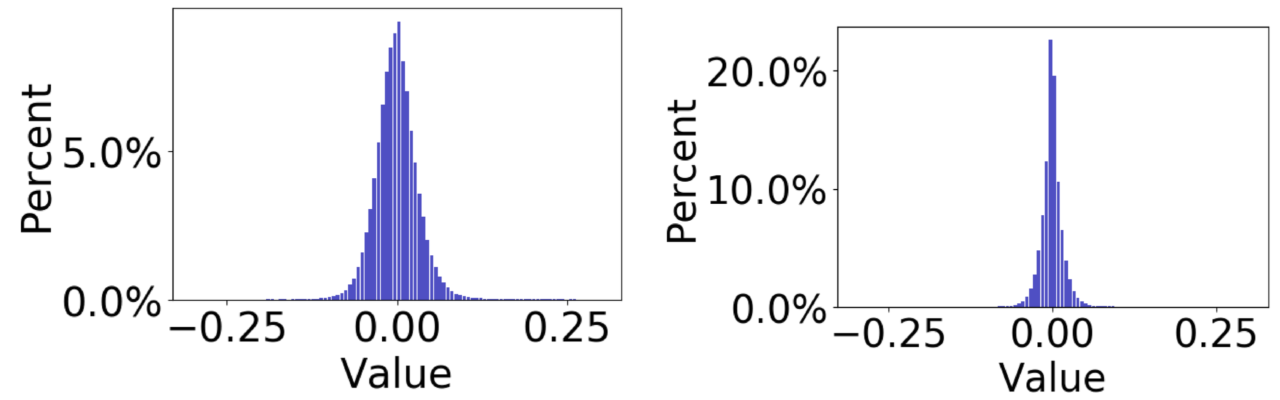
$$d\mathbf{x}(t) = \overline{F}(\mathbf{x}(t), \mathbf{w}(t))dt + \sigma dB_t,$$

and $u(\hat{\mathbf{x}}, 0)$ is the conditional expectation of $f(\mathbf{x}(1))$.

EnResNets – DNN counterpart of $u(\mathbf{x}, 0)$ of the CDE.

1. **Inject noise to each residual mapping of ResNet.**
2. **Average over the output of multiple jointly trained modified ResNet.**

Numerical results - Sparsity of the Weights



Histogram of adversarially trained ResNet20 (L) and En_5ResNet20 (R).

EnResNet has much sparser weights than the ResNet!

Numerical results - Natural & Robust Acc

A_1: natural accuracy
A_2: robust accuracy under FGSM attack
A_3: robust accuracy under IFGSM[20] attack
A_4: robust accuracy under C&W attack

CIFAR10

Model	A_1	A_2	A_3	A_4
ResNet20	75.11%	50.89%	46.03%	58.73%
En_1ResNet20	77.21%	55.35%	49.06%	65.69%
En_2ResNet20	80.34%	57.23%	50.06%	66.47%
En_5ResNet20	82.52%	58.92%	51.48%	67.73%
En_5WideResNet34-10	**86.19%**	**61.82%**	**56.60%**	**69.32%**

CIFAR100

Model	A_1	A_2	A_3	A_4
ResNet20	46.02%	24.77%	23.23%	32.42%
En_2ResNet20	50.68%	30.20%	26.25%	40.06%
En_5ResNet20	51.72%	31.64%	27.80%	40.44%

Conclusion

- Improves DNN's robustness to adversarial attacks
- Improves natural accuracy of the adversarially trained DNNs
- Sparsify the adversarially trained DNNs
- Code at `https://github.com/BaoWangMath/EnResNet`

Ref: B. Wang, B. Yuan, Z. Shi, and S. Osher, ResNets Ensemble via the Feynman-Kac Formalism to Improve Natural and Robust Accuracies, NeurIPS, 2019.