Why Robust Generalization in Deep Learning is Difficult: Perspective of Expressive Power [1,2]

Binghui Li

Peking University

Binghui Li

Jikai Jin

Han Zhong

John Hopcroft

Liwei Wang

---

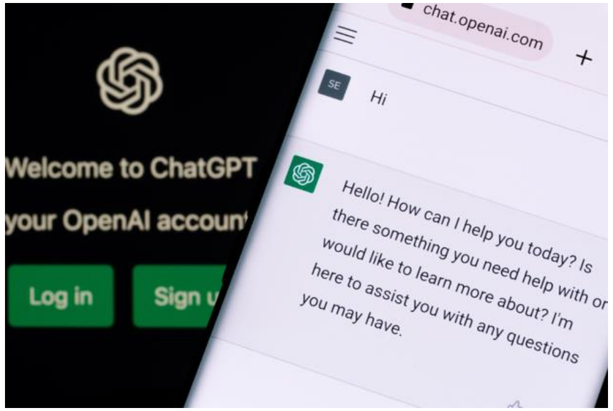
[1] This work has been accepted by **NeurIPS 2022**.

[2] Our full paper can be found at **https://arxiv.org/abs/2205.13863** .

Deep Learning

- Nowadays, deep learning has achieved remarkable success in a variety of disciplines including computer vision, natural language processing, multi-agent decision making as well as scientific and engineering applications.

SAM



ChatGPT

AlphaStar

- Deep Learning ≈ $\underbrace{\text{Deep Neural Network}}_{\text{Powerful Expressivity}} + \underbrace{\text{Gradient Descent}}_{\text{Efficient Opt Alg}}$

Deep Neural Network

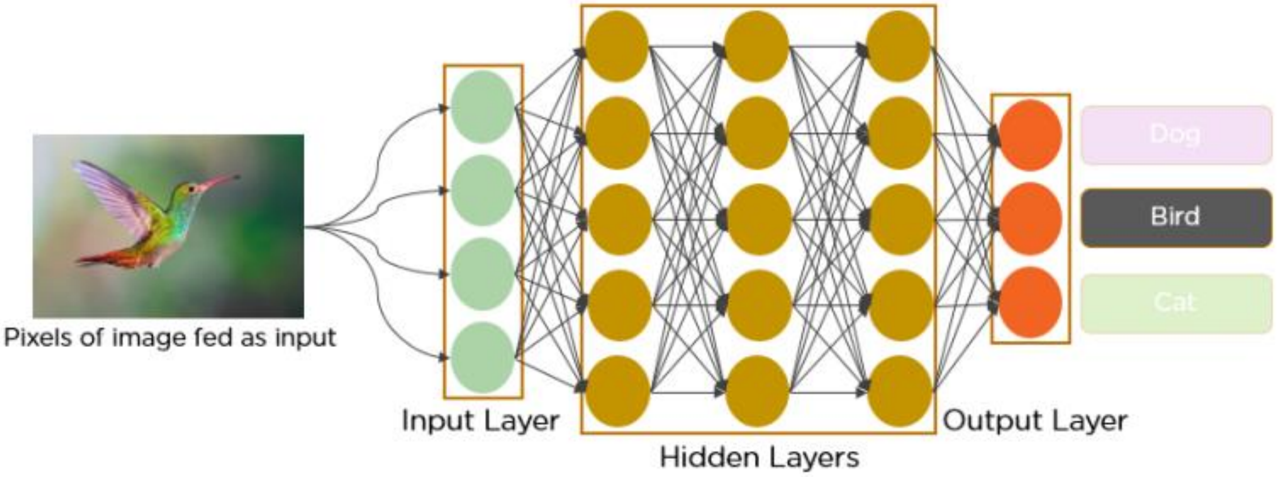
- A multilayer neural network is a function from input $\boldsymbol{x} \in \mathbb{R}^d$ to output $\boldsymbol{y} \in \mathbb{R}^m$, recursively defined as follows:

$$\boldsymbol{h}_1 = \sigma\left(\boldsymbol{W}_1 \boldsymbol{x} + \boldsymbol{b}_1\right), \quad \boldsymbol{W}_1 \in \mathbb{R}^{m_1 \times d}, \boldsymbol{b}_1 \in \mathbb{R}^{m_1},$$

$$\boldsymbol{h}_\ell = \sigma\left(\boldsymbol{W}_\ell \boldsymbol{h}_{\ell-1} + \boldsymbol{b}_\ell\right), \quad \boldsymbol{W}_\ell \in \mathbb{R}^{m_\ell \times m_{\ell-1}}, \boldsymbol{b}_\ell \in \mathbb{R}^{m_\ell}, 2 \leq \ell \leq L-1,$$

$$\boldsymbol{y} = \boldsymbol{W}_L \boldsymbol{h}_L + \boldsymbol{b}_L, \quad \boldsymbol{W}_L \in \mathbb{R}^{m \times m_L}, \boldsymbol{b}_L \in \mathbb{R}^m,$$

where σ is the activation function and L is the depth of the neural network. Here, we mainly focus on ReLU nets i.e. σ(x) = max{0, x}.



Train Deep Model via Gradient Descent

- Data: we consider a binary classification task: $X \to Y \in \{-1, +1\}$, and let D be the data distribution on $X \times Y$.
- Model: parameterized neural network classifier: $\{f_\theta\}_{\theta \in \Theta}$.
- Objective: we evaluate the classification performance by the test loss:

$$L(\theta) = \mathbb{E}_{(x,y)\sim D}[L(f_\theta(x), y)],$$

where $L(\cdot,\cdot)$ denotes loss function, e.g. MSE-loss: $L(z, y) = (z - y)^2$, 0-1loss:$\mathbb{I}\{z \neq y\}$.

In practice, we use empirical average loss on training dataset $S = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ instead of the test loss:

$$\hat{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} L(f_\theta(x_i), y_i).$$

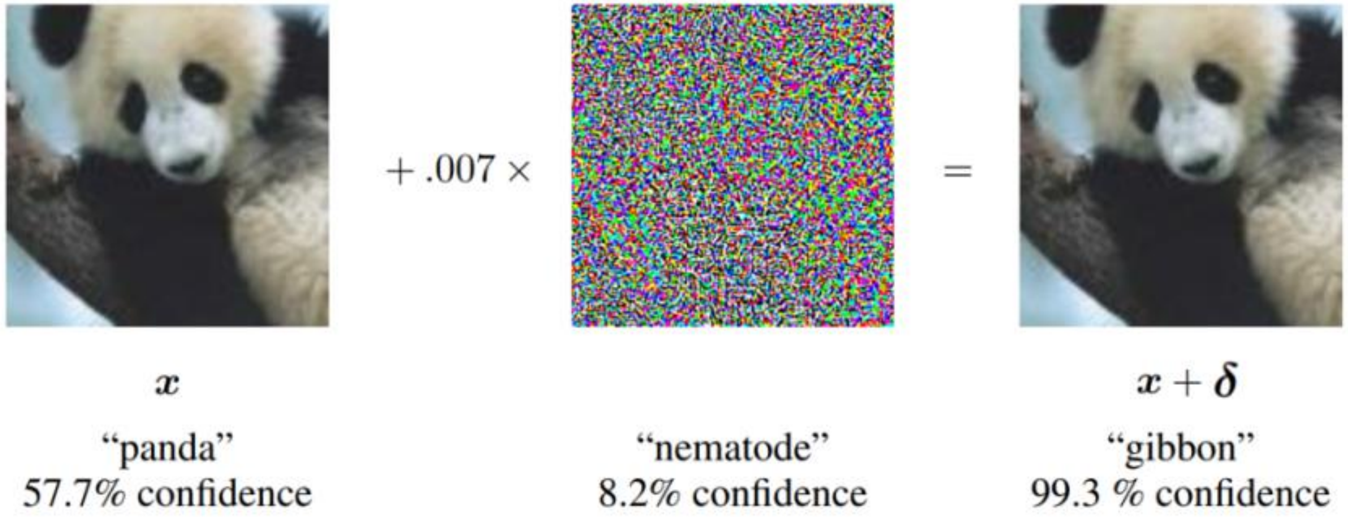
- Training Algorithm: we use gradient descent to minimize the training loss $\hat{L}(\theta)$:

$$\theta \leftarrow \theta - \eta \nabla_\theta \hat{L}(\theta),$$

where η is learning rate.

Adversarial Examples

- Although deep neural networks have achieved remarkable success in practice, it is well-known that modern neural networks are vulnerable to adversarial examples.
- Specifically, for a given image x, an indistinguishable small but adversarial perturbation δ is chosen to fool the classifier f to produce a wrong class using f (x + δ).



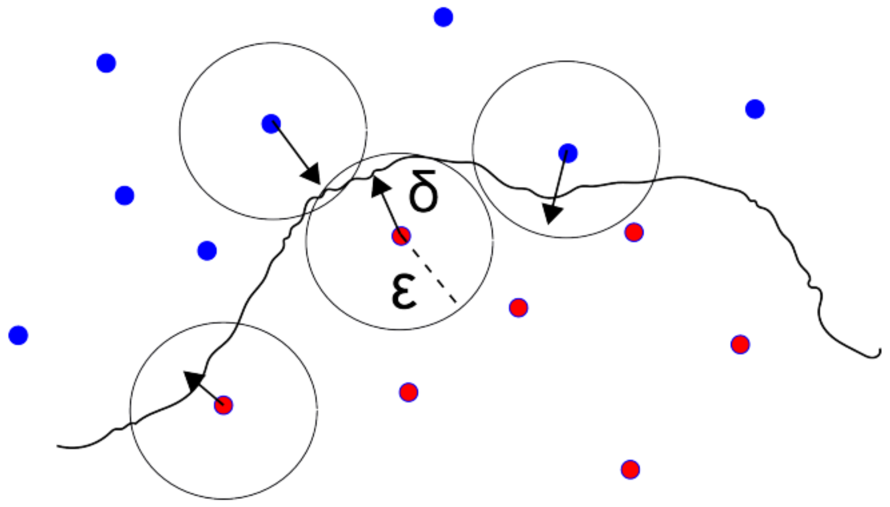
An Instance for Adversarial Example

Adversarial Training

- To mitigate this problem, a common approach is to design adversarial training algorithms by using adversarial examples as training data.

Concretely, we consider a training dataset $S = \{(x_1, y_1), \ldots, (x_N, y_N)\}$, and we aim to solve the following min−max optimization problem：

$$\min_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^{N} \max_{\|\delta\| \leq \varepsilon} L(f_\theta(x_i + \delta), y_i)$$

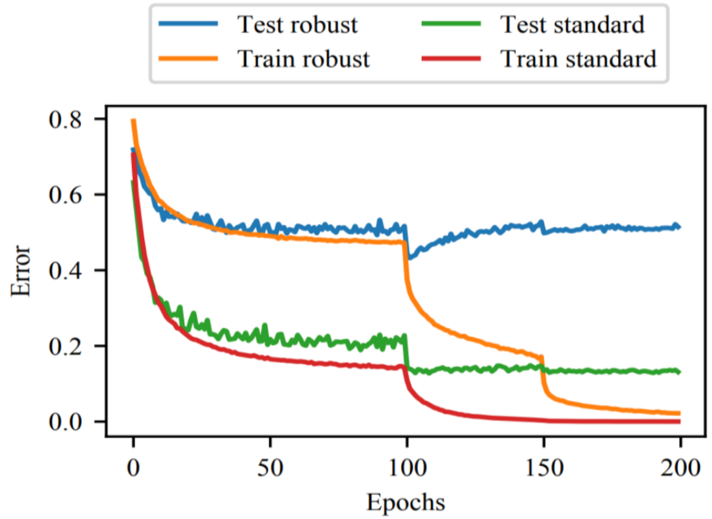


Robust Generalization Gap is Large!

- However, while the state-of-the-art adversarial training methods can achieve high robust training accuracy, all existing methods suffer from large robust test error, which is also called robust overfitting.

	Clean training	Adversarial training
Robust test	3.5%	45.8%
Robust train	–	100%
Clean test	95.2%	87.3%
Clean train	100%	100%

The test and train performance of clean and adversarial training on CIFAR 10 [RXY+19]



The learning curves of adversarial training on CIFAR 10 [RWK20]

Questions

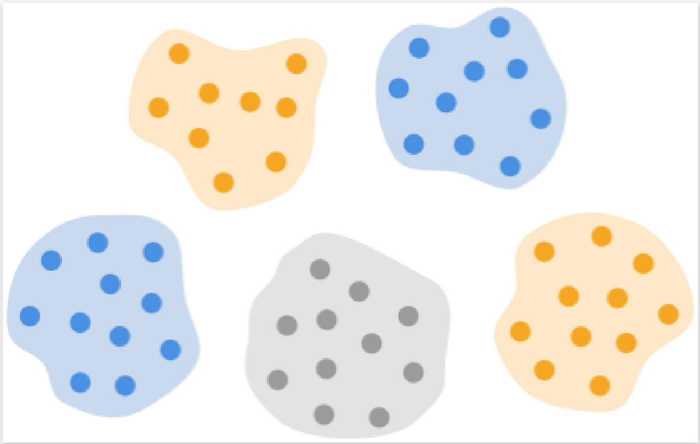
Why does there exist such a ***large generalization gap*** *in the context of robust learning? Can we provide a* ***theoretical understanding*** *of this puzzling phenomena?*

Key Observation

Fact *Data are far from each other.*

	adversarial perturbation ε	minimum Train-Train separation	minimum Test-Train separation
MNIST	0.1	0.737	0.812
CIFAR-10	0.031	0.212	0.220
SVHN	0.031	0.094	0.110
ResImageNet	0.005	0.180	0.224

Experiment results about data separation in [YRZ+20].



Understand Robust Generalization Gap via Representation Complexity

> **Assumption** (Separated Data Distribution)
> *Let D be the binary-labeled data distribution, where data points are in two sets $A, B \subset [0,1]^d$. We assume that separation $d(A, B) \geq 2\epsilon$ and the perturbation radius $\delta < \epsilon$.*

- Representation Complexity:
$$RC(\{f_\theta\}_{\theta \in \Theta}) = \#\, params\ |\theta|$$
- Under the assumption, we focus on:
    - (robust training) *For arbitrary N-size training dataset S i.i.d. sampled from D, how much representation complexity is enough for ReLU nets to achieve **zero robust training error**?*
    - (robust generalization) *For arbitrary data distribution D that satisfies the assumption, how much representation complexity is enough for ReLU nets to achieve **low robust test error**?*

$\tilde{O}(Nd)$ Parameters are Enough to Achieve Zero Robust Training Error

Theorem (Upper Bound for Robust Training)
For any given N-size and d-dim training dataset S that satisfies the separability condition, there exists a ReLU network f with at most $\tilde{O}(Nd)$ parameters such that robust training error is zero.

- For robust training,

$$RC(ReLU\ Nets) = \tilde{O}(Nd).$$

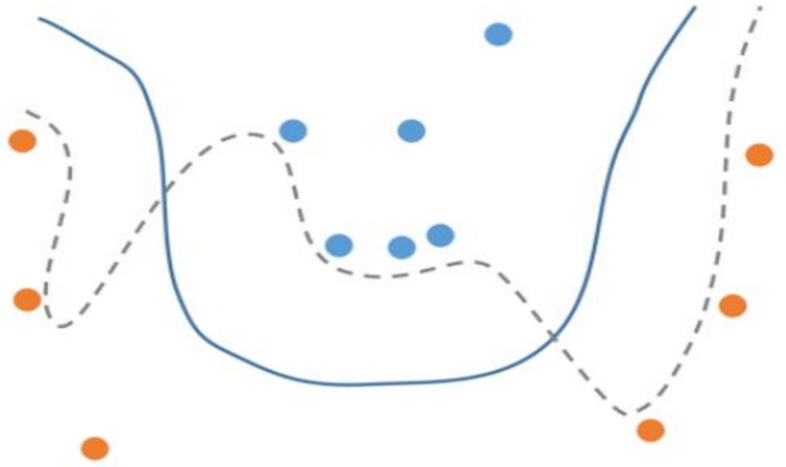
- It is consistent with the common practice that *moderate-size network* trained by adversarial training achieves *high robust training accuracy*.

There Exists a EXP Large Robust Classifier

Lemma

Under the separability assumption, there exists a robust classifier f^* such that it can robustly classify the 2ϵ- separated labeled sets A and B.

- $f^*(x) = \frac{d(x,B)-d(x,A)}{d(x,B)+d(x,A)}$
- f^* is a ϵ^{-1}-Lipschitz function



Theorem

There exists a ReLU net f with at most $O(exp(d))$ params such that $|f - f^*| = o(1)$ for all $x \in [0,1]^d$.

- Corollary: For robust generalization, $RC(ReLU\ Nets) = O(\exp(d))$.

Robust Generalization Requires Exponentially Large Models

- Now, we present our main result in this paper.

Theorem (Lower Bound for Robust Generalization)
Let F_m be the family of function represented by ReLU nets with at most m parameters. Then, there exists a number $m(d) = \Omega(exp(d))$ and a linear-separable distribution D satisfying the assumption such that, for any classifier in $F_{m(d)}$, the robust test error is at least $\Omega(1)$.

- For robust generalization,

$$RC(ReLU\ Nets) = \Omega(\exp(d)),$$

in contrast, for standard generalization, only $O(d)$ params are enough.

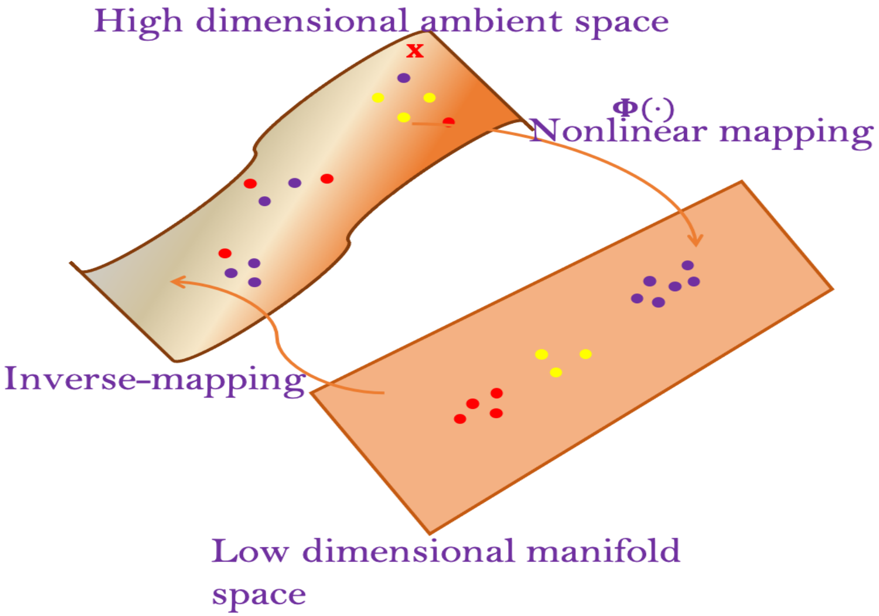
- Moreover, this lower bound holds for *arbitrarily small* perturbation radius and *general models* as long as $VCDim = O(poly(\#params))$.

Robust Generalization for Low-dimensional-manifold Data

- A common belief of real-life data such as images is that the data points lie on a **low-dimensional manifold**.

Assumption (Manifold Data)
We assume that data lies on a manifold M with the intrinsic dimension k ($k \ll d$), where data points are in two separated labeled sets $A, B \subset M$.

Theorem (Improved Upper Bound)
Under the manifold-data assumption, there exists a ReLU net with at most $\tilde{O}(\exp(k))$ params such that the robust test error on the manifold is zero.



Conclusion

Take Home Message: From the view of representation complexity,
(1) Robust training only needs linearly large models;
(2) Robust generalization, in worst case, requires EXP larger models.

Table 1: Summary of our main results.

Params	Setting			
	Robust Training	Robust Generalization		
		General Case	Linear Separable	$k-$dim Manifold
Upper Bound	$\mathcal{O}(Nd)$ (Thm 2.2)	$\exp(\mathcal{O}(d))$ (Thm 3.3)		$\exp(\mathcal{O}(k))$ (Thm 5.5)
Lower Bound	$\Omega(\sqrt{Nd})$ (Thm 2.3)	$\exp(\Omega(d))$ (Thm 3.4)	$\exp(\Omega(d))$ (Thm 4.3)	$\exp(\Omega(k))$ (Thm 5.8)

Thanks for listening!

Our full paper can be found at:
https://arxiv.org/abs/2205.13863

My Homepage

This Paper

Recent Related Paper