Mathematical Theory of Adversarial Deep Learning

Xiao-Shan Gao

Academy of Mathematics and Systems Science
Chinese Academy of Sciences

Table of Contents

1. Introduction to Adversarial Deep Learning
2. Robust Memorization: Existence of Robust DNNs
3. Achieving Optimal Robustness via Stackelberg Game
4. Information-theoretically Safe Bias Classifier
5. Summary

Deep Neural Network (DNN)

DNN is the central tool in the current AI breakthroughs:

- Computer Vision
- Natural Language Translation
- Game Playing: AlphaGo
- Autonomous Driving: Vision and Decision
- Protein Structure Prediction: AlphaFold
- and applications in almost every area

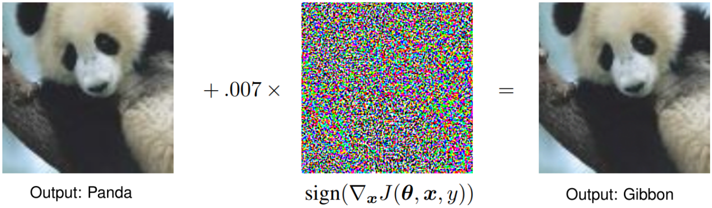
Problems with DNNs

Trade the rigourous for representation power:

- **Robustness and Safety**
- Explainability and causality/reasoning
- Transferability and catastrophic forgetting
- Dependence too much on large amount of data and computation
- Lack of rigourous and applicable theory for training and generalization

Adversarial Samples and Adversarial Attack

With little modifications which are essentially imperceptible to the human eye, DNN outputs a wrong label

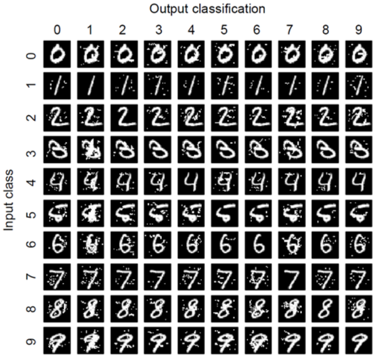


Adversarial Attack   Adversarial Samples

(Goodfellow-Shlens-Szegedy, 2014)

Targeted Adversary Attack

With little modifications, DNN outputs any label given by the adversary

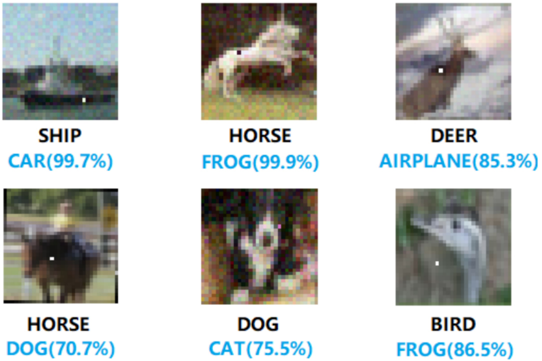


Modify 4% **pixels of MNIST:** 97% images have adversaries

(Papernot et al. 2016)

Single-pixel Adversary Attack

Modify a Single Pixel: 67% of CIFAR-10 have adversaries



(Su-Vargas-Sakurai, 2019)

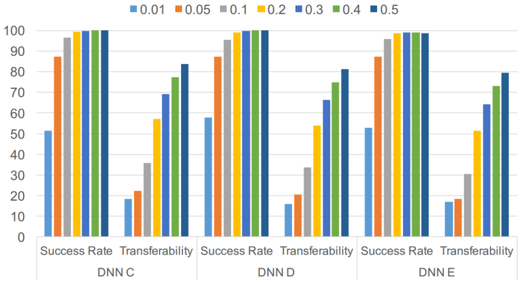
White-box vs Black-box Adversary Attacks

White-box Attack: the parameters of the DNN are known and the gradients of the DNN are used to generate adversaries.

White-box vs Black-box Adversary Attacks

White-box Attack: the parameters of the DNN are known and the gradients of the DNN are used to generate adversaries.

Black-box Attack: Based on transferability of adversarial examples: An adversary of $\mathcal{C}_1$ is likely to be the adversary for a “similar” $\mathcal{C}_2$.



(Papernot et al, 2016)

Defence against White-box Attack: Gradient Masking

Hides the gradient to avoid gradient based white-box attacks:

- Let $\mathcal{G}(x)$ be a "small" step function or random function, which does not have meaningful gradient.
- Use $\mathcal{G}(x)$ instead of x as the input.

Defence against White-box Attack: Gradient Masking

Hides the gradient to avoid gradient based white-box attacks:

- Let $\mathcal{G}(x)$ be a "small" step function or random function, which does not have meaningful gradient.
- Use $\mathcal{G}(x)$ instead of x as the input.

Defence does not work: Local minor changes can be recovered!

Gradient Masking	Successful Attack
Shattered Gradients	Approximate the step function
Stochastic Gradients	Compute the expectation
Vanishing Gradients	Reparameterization

(Athalye-Carlini-Wagner, 2018)

Adversarial Training for More Robust DNNs

Normal Training: $\Theta^* = \arg\min_{\Theta \in \mathbb{R}^K} \mathbb{E}_{(x,y)\sim\mathcal{D}} \mathrm{Loss}(\mathcal{C}_\Theta(x), y)$

Adversarial Training for More Robust DNNs

Normal Training: $\Theta^* = \arg\min_{\Theta \in \mathbb{R}^K} \mathbb{E}_{(x,y)\sim\mathcal{D}} \mathrm{Loss}(\mathcal{C}_\Theta(x), y)$

Adversarial Training (Madry et al, 2017)

Given an attack radius $\varepsilon \in \mathbb{R}_+$, AT is a robust optimization problem:

$$\Theta^* = \arg\min_{\Theta \in \mathbb{R}^K} \mathbb{E}_{(x,y)\sim\mathcal{D}} \max_{||\overline{x}-x|| \le \varepsilon} \mathrm{Loss}(\mathcal{C}_\Theta(\overline{x}), y)$$

Empirical risk minimization over the most-adversarial sample $\overline{x}$ of x

Adversarial Training for More Robust DNNs

Normal Training: $\Theta^* = \arg\min_{\Theta \in \mathbb{R}^K} \mathbb{E}_{(x,y)\sim\mathcal{D}} \mathrm{Loss}(\mathcal{C}_\Theta(x), y)$

Adversarial Training (Madry et al, 2017)

Given an attack radius $\varepsilon \in \mathbb{R}_+$, AT is a robust optimization problem:

$$\Theta^* = \arg\min_{\Theta \in \mathbb{R}^K} \mathbb{E}_{(x,y)\sim\mathcal{D}} \max_{||\overline{x}-x||\le\varepsilon} \mathrm{Loss}(\mathcal{C}_\Theta(\overline{x}), y)$$

Empirical risk minimization over the most-adversarial sample $\overline{x}$ of x

Adversarial training is the best empirical defence

	ϵ	Without AT		With AT	
		Accuracy	Adv. Accu	Accuracy	Adv. Accu
MNIST	0.1	99%	76%	99%	97%
CIFAR10	0.03	90%	0%	83%	49%

Adversarial Accuracy: Percentage of samples without adversarial examples

Adversarial Samples are Inevitable!

- State-of-the-art DNNs have adversaries.

Adversarial Samples are Inevitable!

- State-of-the-art DNNs have adversaries.
- For any given DNN $\mathcal{C}$, $\exists\mathcal{D}$ such that if $\mathcal{C}$ is accurate on $\mathcal{D}$, then $\mathcal{C}$ has adversaries over $\mathcal{D}$ with high probabilty. (Bastounis et al, 2020)

Adversarial Samples are Inevitable!

- State-of-the-art DNNs have adversaries.
- For any given DNN $\mathcal{C}$, $\exists\mathcal{D}$ such that if $\mathcal{C}$ is accurate on $\mathcal{D}$, then $\mathcal{C}$ has adversaries over $\mathcal{D}$ with high probabilty. (Bastounis et al, 2020)
- DNN is also extremely sensitive to its parameters:
  If the width of a DNN $\mathcal{C}$ is sufficiently large,
  then we can change the parameters of $\mathcal{C}$ as small as possible,
  such that the modified DNN has adversarial samples as close as possible to the normal samples. (Yu-Wang-Gao, 2022)

Adversarial Samples are Inevitable!

- State-of-the-art DNNs have adversaries.
- For any given DNN $\mathcal{C}$, $\exists\mathcal{D}$ such that if $\mathcal{C}$ is accurate on $\mathcal{D}$, then $\mathcal{C}$ has adversaries over $\mathcal{D}$ with high probabilty. (Bastounis et al, 2020)
- DNN is also extremely sensitive to its parameters:
  If the width of a DNN $\mathcal{C}$ is sufficiently large,
  then we can change the parameters of $\mathcal{C}$ as small as possible,
  such that the modified DNN has adversarial samples as close as possible to the normal samples. (Yu-Wang-Gao, 2022)

Adversary is a key factor for safety-critical applications, such as autonomous driving, financial authentication, military camouflage

Some Basic Issues of Adversarial Learning

Adversarial Learning: Learning at the existence of adversaries

Some Basic Issues of Adversarial Learning

Adversarial Learning: Learning at the existence of adversaries

- Does there exists a robust classifier against any adversarial attack?

Some Basic Issues of Adversarial Learning

Adversarial Learning: Learning at the existence of adversaries

- Does there exists a robust classifier against any adversarial attack?
- How to train a classifier from a given hypothesis space, which ensures optimal robustness against any adversarial attack?

Some Basic Issues of Adversarial Learning

Adversarial Learning: Learning at the existence of adversaries

- Does there exists a robust classifier against any adversarial attack?
- How to train a classifier from a given hypothesis space, which ensures optimal robustness against any adversarial attack?
- Does there exist provable adversarial robust and practical classifiers?
- $\cdots$

DNN: Piecewise Linear and Continuous Function

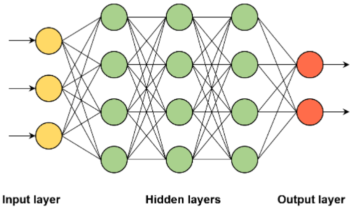
DNN: $\mathcal{C} : [0,1]^d \to \mathbb{R}^m$

***l*-th Hidden Layer:**

$$x_l = \mathrm{Relu}(W_l x_{l-1} + b_l)$$

$$\mathrm{Relu}(x) = \max\{0, x\}$$

Parameters of $\mathcal{C}$: $\Theta = \{W_l, b_l\}_{l=1}^{L}$



DNN: Piecewise Linear and Continuous Function

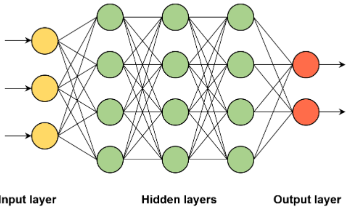
DNN: $\mathcal{C} : [0,1]^d \to \mathbb{R}^m$

***l*-th Hidden Layer:**

$x_l = \mathrm{Relu}(W_l x_{l-1} + b_l)$

$\mathrm{Relu}(x) = \max\{0, x\}$

Parameters of $\mathcal{C}$: $\Theta = \{W_l, b_l\}_{l=1}^{L}$



Training:

Given a data set: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$

Empirical risk minimization

$\Theta^* = \arg\min_\Theta \sum_i \mathrm{Loss}(\mathcal{C}_\Theta(x_i), y_i)$

DNN: Piecewise Linear and Continuous Function

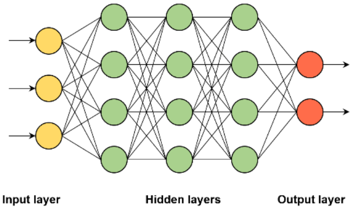
DNN: $\mathcal{C} : [0,1]^d \to \mathbb{R}^m$

***l*-th Hidden Layer:**

$x_l = \mathrm{Relu}(W_l x_{l-1} + b_l)$

$\mathrm{Relu}(x) = \max\{0, x\}$

Parameters of $\mathcal{C}$: $\Theta = \{W_l, b_l\}_{l=1}^L$

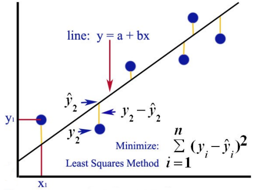


Training:

Given a data set: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$

Empirical risk minimization

$\Theta^* = \arg\min_\Theta \sum_i \mathrm{Loss}(\mathcal{C}_\Theta(x_i), y_i)$



DNN: Piecewise Linear and Continuous Function

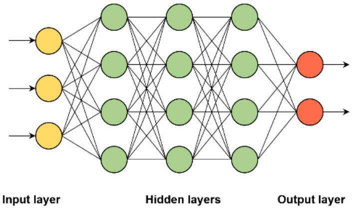
DNN: $\mathcal{C} : [0,1]^d \to \mathbb{R}^m$

***l*-th Hidden Layer:**

$x_l = \mathrm{Relu}(W_l x_{l-1} + b_l)$

$\mathrm{Relu}(x) = \max\{0, x\}$

Parameters of $\mathcal{C}$: $\Theta = \{W_l, b_l\}_{l=1}^L$

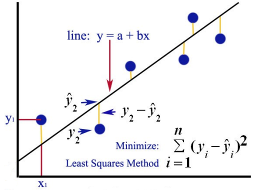


Training:

Given a data set: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$

Empirical risk minimization

$\Theta^* = \arg\min_\Theta \sum_i \mathrm{Loss}(\mathcal{C}_\Theta(x_i), y_i)$



Deep Learning: Approximate a high dimensional ($d \sim 784 - 150528$) function with a piecewise continuous linear function

Classification DNN

MNIST: Ten hand-written numbers

$d = 28 \cdot 28 = 784$

CIFAR10: Ten objects

$d = 32 \cdot 32 \cdot 3 = 3072$

Classification DNN

MNIST: Ten hand-written numbers

$d = 28 \cdot 28 = 784$

CIFAR10: Ten objects

$d = 32 \cdot 32 \cdot 3 = 3072$

Classification DNN: $\mathcal{C} : [0,1]^d \to \mathbb{R}^{10}$

The Classification Result: $\widehat{\mathcal{C}}(x) = \arg\max_{l=1}^{10} C_l(x)$

Robust Memorization: Existence of Robust DNNs

Data Separation Bound

Data Set: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N} \subset \mathbb{R}^d \times [m]$, where $[m] = \{i\}_{i=1}^{m}$

Separation Bound for $\mathcal{D}$:

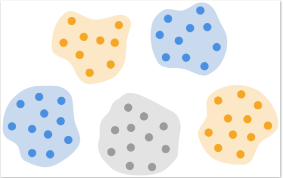
$$\lambda(\mathcal{D}) = \min\{||x_i - x_j||_\infty \,|\, (x_i, y_i), (x_j, y_j) \in \mathcal{D} \text{ and } y_i \neq y_j\}.$$

Data Set: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N \subset \mathbb{R}^d \times [m]$, where $[m] = \{i\}_{i=1}^m$

Separation Bound for $\mathcal{D}$:

$$\lambda(\mathcal{D}) = \min\{||x_i - x_j||_\infty \mid (x_i, y_i), (x_j, y_j) \in \mathcal{D} \text{ and } y_i \neq y_j\}.$$

Sep-Bound	Attack-R	Tr-Tr	Tr-Te
MNIST	0.10	0.73	0.81
CIFAR10	0.03	0.21	0.22
TImageNet	0.005	0.18	0.22



The separation bounds are $\gg$ the usually used attack radii.

Memorization and Robust Memorization

Data Set: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N} \subset \mathbb{R}^d \times [m]$ with separation bound $\lambda(\mathcal{D})$

$\mathcal{C} : \mathbb{R}^d \to \mathbb{R}$ is a memorization DNN of $\mathcal{D}$, if $\mathcal{C}(x_i) = y_i, \forall i \in [N]$

Memorization network exists:

- With depth 2 and width $O(N)$ (Zhang et al, 2017)
- With width 12 and depth $\widetilde{O}(\sqrt{N})$ for separated data (Vardi et al, 2021)

Memorization and Robust Memorization

Data Set: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N \subset \mathbb{R}^d \times [m]$ with separation bound $\lambda(\mathcal{D})$

$\mathcal{C} : \mathbb{R}^d \to \mathbb{R}$ is a memorization DNN of $\mathcal{D}$, if $\mathcal{C}(x_i) = y_i, \forall i \in [N]$

Memorization network exists:

- With depth 2 and width $O(N)$ (Zhang et al, 2017)
- With width 12 and depth $\widetilde{O}(\sqrt{N})$ for separated data (Vardi et al, 2021)

Robust Memorization with a network $\mathcal{C} : \mathbb{R}^d \to \mathbb{R}$

- **Robust Memorization with radius** μ:

  $$\mathcal{C}(x) = y_i \text{ for all } ||x - x_i|| \leq \mu.$$

- **Optimal Robust Memorization**: $\mathcal{C}$ is robust for all $\mu < \lambda(\mathcal{D})/2$.

Robust Memorization is Harder than Memorization

Data Set: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N \subset \mathbb{R}^d \times [m]$

Memorization with a network $\mathcal{C} : \mathbb{R}^d \to \mathbb{R}$, if $\mathcal{C}(x_i) = y_i, \forall i \in [N]$

Memorization Networks exist:

- With depth 2 and width $O(N)$ (Zhang et al, 17)
- With width 12 and depth $\widetilde{O}(\sqrt{N})$ (Vardi et al, 21)

Robust Memorization is Harder than Memorization

Data Set: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N} \subset \mathbb{R}^d \times [m]$

Memorization with a network $\mathcal{C} : \mathbb{R}^d \to \mathbb{R}$, if $\mathcal{C}(x_i) = y_i, \forall i \in [N]$

Memorization Networks exist:

- With depth 2 and width $O(N)$ (Zhang et al, 17)
- With width 12 and depth $\widetilde{O}(\sqrt{N})$ (Vardi et al, 21)

Robust memorization is more difficult than memorization:

- If $\mathcal{C}$ is of depth 2, then there exists a data set $\mathcal{D}$ such that $\mathcal{C}$ is not an optimal robust memorization of $\mathcal{D}$.

Robust Memorization is Harder than Memorization

Data Set: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N \subset \mathbb{R}^d \times [m]$

Memorization with a network $\mathcal{C} : \mathbb{R}^d \to \mathbb{R}$, if $\mathcal{C}(x_i) = y_i, \forall i \in [N]$

Memorization Networks exist:

- With depth 2 and width $O(N)$ (Zhang et al, 17)
- With width 12 and depth $\widetilde{O}(\sqrt{N})$ (Vardi et al, 21)

Robust memorization is more difficult than memorization:

- If $\mathcal{C}$ is of depth 2, then there exists a data set $\mathcal{D}$ such that $\mathcal{C}$ is not an optimal robust memorization of $\mathcal{D}$.
- If $\mathcal{C}$ is of fixed width, then there exists a data set $\mathcal{D}$ such that $\mathcal{C}$ is not an optimal robust memorization of $\mathcal{D}$.

Optimal Robust Memorization with a DNN

Data: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N} \subset \mathbb{R}^d \times [m]$ with separation bound $2 = \lambda(\mathcal{D})$

Robust Classifiers Exist:

- $F(x) = (y_1 + \|x - \mathcal{X}_1\|, \ldots, y_m + \|x - \mathcal{X}_m\|)$ is optimal-robust, because it is 1-Lipschitz (Yang et al, 2020)
- A robust DNN exists due to the universal approximation power of DNN (Bastounis et al, 2020), (Liang-Huang, 2021)

But, the structure (depth/width) of the DNN is not given.

Optimal Robust Memorization with a DNN

Data: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N \subset \mathbb{R}^d \times [m]$ with separation bound $2 = \lambda(\mathcal{D})$

Robust Classifiers Exist:

- $F(x) = (y_1 + \|x - \mathcal{X}_1\|, \ldots, y_m + \|x - \mathcal{X}_m\|)$ is optimal-robust, because it is 1-Lipschitz (Yang et al, 2020)
- A robust DNN exists due to the universal approximation power of DNN (Bastounis et al, 2020), (Liang-Huang, 2021)

But, the structure (depth/width) of the DNN is not given.

Theorem (Effective Memorization. Yu-Gao, 2022)

The set of DNNs with width $O(d)$ and depth $O(N)$ provides an optimal robust memorization for $\mathcal{D}$.

Optimal Robust Memorization with a DNN

Data: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N \subset \mathbb{R}^d \times [m]$ with separation bound $2 = \lambda(\mathcal{D})$

Robust Classifiers Exist:

- $F(x) = (y_1 + \|x - \mathcal{X}_1\|, \ldots, y_m + \|x - \mathcal{X}_m\|)$ is optimal-robust, because it is 1-Lipschitz (Yang et al, 2020)
- A robust DNN exists due to the universal approximation power of DNN (Bastounis et al, 2020), (Liang-Huang, 2021)

But, the structure (depth/width) of the DNN is not given.

Theorem (Effective Memorization. Yu-Gao, 2022)

The set of DNNs with width $O(d)$ and depth $O(N)$ provides an optimal robust memorization for $\mathcal{D}$.

Compare: Approximate general functions needs exponential (3^d) width!

Robust Memorization via Controlling Lipschitz

Achieving robustness by controlling Lipschitz is widely studied.

Using Lipschitz is potentially harder:

There exists a data set: $\mathcal{T} = \{(x_i, y_i)\}_{i=0}^{d} \subset \mathbb{R}^d \times \{-1, 1\}$, with $\lambda(\mathcal{D}) = 1$

- Optimal robust memorization exists: with depth 2 and width $2d$
- Networks with depth 2 cannot be optimal robust mem. for $\mathcal{T}$ via Lipschitz

Robust Memorization via Controlling Lipschitz

Achieving robustness by controlling Lipschitz is widely studied.

Using Lipschitz is potentially harder:

There exists a data set: $\mathcal{T} = \{(x_i, y_i)\}_{i=0}^{d} \subset \mathbb{R}^d \times \{-1, 1\}$, with $\lambda(\mathcal{D}) = 1$

- Optimal robust memorization exists: with depth 2 and width $2d$
- Networks with depth 2 cannot be optimal robust mem. for $\mathcal{T}$ via Lipschitz

Theorem (Yu-Gao, 2022)

There exists a network with width $O(d)$ and depth $O(N\log(d))$, which is an optimal robust memorization for $\mathcal{D}$ via Lipschitz.

Comparing width $O(d)$ and depth $O(N)$ without using Lipschitz.

Summary on the Existence of Robust DNNs

For a data set: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N \subset \mathbb{R}^d \times [m]$

- Optimal robust DNNs width $O(d)$ and depth $O(N)$ exist and can be computed in polynomial time.

  But, the depth ($N > 60000$) is too big to be practical.

Summary on the Existence of Robust DNNs

For a data set: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N \subset \mathbb{R}^d \times [m]$

- Optimal robust DNNs width $O(d)$ and depth $O(N)$ exist and can be computed in polynomial time.

  But, the depth ($N > 60000$) is too big to be practical.

- Finding robust DNNs with one hidden layer and width 2 is NP-hard (Yu-Gao, 2022).

Summary on the Existence of Robust DNNs

For a data set: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N \subset \mathbb{R}^d \times [m]$

- Optimal robust DNNs width $O(d)$ and depth $O(N)$ exist and can be computed in polynomial time.

  But, the depth ($N > 60000$) is too big to be practical.

- Finding robust DNNs with one hidden layer and width 2 is NP-hard (Yu-Gao, 2022).

- In between, we may ask

  For DNNs with given fixed depth and width, how to achieve the optimal robustness?

Achieving Optimal Robustness via Stackelberg Game

Adversarial Learning as a Game

Player 1: Classifier:

- **Strategy Space:** $\mathcal{S}_c = [-E, E]^K$

To compute robust DNN with parameters $\Theta \in \mathcal{S}_c$: $\mathcal{C}_\Theta : \mathbb{I}^d \to \mathbb{R}^m$.

Adversarial Learning as a Game

Player 1: Classifier:

- **Strategy Space:** $\mathcal{S}_c = [-E, E]^K$

To compute robust DNN with parameters $\Theta \in \mathcal{S}_c$: $\mathcal{C}_\Theta : \mathbb{I}^d \to \mathbb{R}^m$.

Player 2: Adversary:

- **Strategy Space**: $\mathcal{S}_a = \{A : \mathcal{X} \to \mathbb{B}_\varepsilon\}$, where $\mathbb{B}_\varepsilon = \{\delta \in \mathbb{R}^d \ : \ ||\delta|| \leq \varepsilon\}$

To compute Adversarial Sample: $x + A(x)$ for x.

Adversarial Learning as a Game

Player 1: Classifier:

- **Strategy Space:** $\mathcal{S}_c = [-E, E]^K$

To compute robust DNN with parameters $\Theta \in \mathcal{S}_c$: $\mathcal{C}_\Theta : \mathbb{I}^d \to \mathbb{R}^m$.

Player 2: Adversary:

- **Strategy Space**: $\mathcal{S}_a = \{A : \mathcal{X} \to \mathbb{B}_\varepsilon\}$, where $\mathbb{B}_\varepsilon = \{\delta \in \mathbb{R}^d \ : \ ||\delta|| \le \varepsilon\}$

To compute Adversarial Sample: $x + A(x)$ for x.

A two-player zero-sum game:

Payoff function: $\phi(\Theta, A) = \mathbb{E}_{(x,y)\sim\mathcal{D}} \operatorname{Loss}(\mathcal{C}_\Theta(x + A(x)), y)$

Goals of the players:

Classifier: $\min_{\Theta\in\mathcal{S}_c} \phi(\Theta, A)$

Adversary: $\max_{A\in\mathcal{S}_a} \phi(\Theta, A)$

Nash Equilibrium of the Adversarial Game

Nash Equilibrium: $(\Theta^*, A^*) \in \mathcal{S}_c \times \mathcal{S}_a$

$$\phi(\Theta^*, A^*) \leq \phi(\Theta, A^*) \quad \text{and} \quad \phi(\Theta^*, A^*) \geq \phi(\Theta^*, A)$$

At Nash Equilibrium, no player can benefit by unilaterally changing its strategy, so it gives an optimal defence against adversarial attacks.

Nash Equilibrium of the Adversarial Game

Nash Equilibrium: $(\Theta^*, A^*) \in \mathcal{S}_c \times \mathcal{S}_a$

$$\phi(\Theta^*, A^*) \leq \phi(\Theta, A^*) \quad \text{and} \quad \phi(\Theta^*, A^*) \geq \phi(\Theta^*, A)$$

At Nash Equilibrium, no player can benefit by unilaterally changing its strategy, so it gives an optimal defence against adversarial attacks.

Nash Equilibrium does not exist for DNNs!

Nash Equilibrium of the Adversarial Game

Nash Equilibrium: $(\Theta^*, A^*) \in \mathcal{S}_c \times \mathcal{S}_a$

$\phi(\Theta^*, A^*) \leq \phi(\Theta, A^*)$ and $\phi(\Theta^*, A^*) \geq \phi(\Theta^*, A)$

At Nash Equilibrium, no player can benefit by unilaterally changing its strategy, so it gives an optimal defence against adversarial attacks.

Nash Equilibrium does not exist for DNNs!

Nash Equilibrium exists if

- $\mathcal{S}_c$ is convex and $\mathcal{S}_a$ is prob distributions (Bose et al, 2020)
- $\mathcal{S}_c$ and $\mathcal{S}_a$ are parameterized by prob distributions (Gidel et al, 2020)
- **Mixed Nash Equilibrium:** Probability distributions over $\mathcal{S}_c$ and $\mathcal{S}_a$

Not answer the question of optimal robustness for DNNs with fixed structure.

Adversarial Learning as a Stackelberg Game

A zero-sum Stackelberg game: $\min_{\Theta\in\mathcal{S}_c} \max_{A\in\mathcal{S}_a} \phi(\Theta, A)$

- **Classifier plays first:** $\min_{\Theta\in\mathcal{S}_c} \phi(\Theta, A)$, knowing the Adversary
- **Adversary play subsequently knowing the decision of the Classifier:** $\max_{A\in\mathcal{S}_a} \phi(\Theta, A)$

Adversarial Learning as a Stackelberg Game

A zero-sum Stackelberg game: $\min_{\Theta\in\mathcal{S}_c} \max_{A\in\mathcal{S}_a} \phi(\Theta, A)$

- **Classifier plays first:** $\min_{\Theta\in\mathcal{S}_c} \phi(\Theta, A)$, knowing the Adversary
- **Adversary play subsequently knowing the decision of the Classifier:** $\max_{A\in\mathcal{S}_a} \phi(\Theta, A)$

Stackelberg Equilibrium: $(\Theta^*, A^*) \in \mathcal{S}_c \times \mathcal{S}_a$

- $A(\Theta) = \arg\max_{A\in\mathcal{S}_A} \phi(\Theta, A)$ exists for any $\Theta \in \mathcal{S}_c$, and
- $\Theta^* \in \arg\min_{\Theta\in\mathcal{S}_c} \phi(\Theta, A(\Theta))$ and $A^* = A(\Theta^*)$

Adversarial Learning as a Stackelberg Game

A zero-sum Stackelberg game: $\min_{\Theta\in\mathcal{S}_c} \max_{A\in\mathcal{S}_a} \phi(\Theta, A)$

- **Classifier plays first:** $\min_{\Theta\in\mathcal{S}_c} \phi(\Theta, A)$, knowing the Adversary
- **Adversary play subsequently knowing the decision of the Classifier:** $\max_{A\in\mathcal{S}_a} \phi(\Theta, A)$

Stackelberg Equilibrium: $(\Theta^*, A^*) \in \mathcal{S}_c \times \mathcal{S}_a$

- $A(\Theta) = \arg\max_{A\in\mathcal{S}_A} \phi(\Theta, A)$ exists for any $\Theta \in \mathcal{S}_c$, and
- $\Theta^* \in \arg\min_{\Theta\in\mathcal{S}_c} \phi(\Theta, A(\Theta))$ and $A^* = A(\Theta^*)$

Stackelberg equilibrium exists if the strategy spaces are compact and the payoff function is continuous (Simaan-Cruz, 1973)

In our case, $\mathcal{S}_c$ is compact, but $\mathcal{S}_a$ is not.

Stackelberg Equilibrium Exists

Theorem (Gao-Liu-Yu, 2022)

Game G has a Stackelberg equilibrium (Θ^*, A^*).
Θ^* *is the solution to the adversarial training (Madry et al, 17).*

Key Observation:

Although $\mathcal{S}_a = \{A : \mathbb{I}^d \to \mathbb{B}_\varepsilon\}$ is not compact
$\mathbb{B}_\varepsilon = \{\delta \in \mathbb{R}^d \, : \, ||\delta|| \le \varepsilon\}$ is compact. Thus

$$\begin{aligned} A(\Theta) &= \arg\max_{A \in \mathcal{S}_A} \phi(\Theta, A) \text{ iff} \\ A(\Theta)(x) &= \arg\max_{A(x) \in \mathbb{B}_\varepsilon} \mathrm{Loss}(\mathcal{C}_\Theta(x + A(x)), y) \end{aligned}$$

Stackelberg Equilibrium Exists

Theorem (Gao-Liu-Yu, 2022)

Game G has a Stackelberg equilibrium (Θ^*, A^*).
Θ^* *is the solution to the adversarial training (Madry et al, 17).*

Key Observation:

Although $\mathcal{S}_a = \{A : \mathbb{I}^d \to \mathbb{B}_\varepsilon\}$ is not compact
$\mathbb{B}_\varepsilon = \{\delta \in \mathbb{R}^d \; : \; ||\delta|| \le \varepsilon\}$ is compact. Thus

$$\begin{aligned} A(\Theta) &= \arg\max_{A \in \mathcal{S}_A} \phi(\Theta, A) \text{ iff} \\ A(\Theta)(x) &= \arg\max_{A(x) \in \mathbb{B}_\varepsilon} \mathrm{Loss}(\mathcal{C}_\Theta(x + A(x)), y) \end{aligned}$$

The solution gives optimal adversarial empirical risk:

$$\mathbb{E}_{(x,y)\sim\mathcal{D}} \max_{||\overline{x}-x|| \le \varepsilon} \mathrm{Loss}(\mathcal{C}_{\Theta^*}(\overline{x}), y)$$

which depends on the loss function Loss and is not intrinsic.

The Optimal Robust Classifier

Adversarial Accuracy of a DNN $\mathcal{C}$ wrt an attack radius ε: intrinsic robustness measurement.

$$\mathrm{AA}_{\mathcal{D}}(\mathcal{C},\varepsilon) = \mathbb{P}_{(x,y)\sim\mathcal{D}}\left(\forall \overline{x} \in \mathbb{B}(x,\varepsilon)\,(\widehat{\mathcal{C}}(\overline{x}) = y)\right)$$

Carlini-Wagner loss function: $\mathrm{Loss}_{\mathrm{cw}}(z,y) = \max_{l\in[m],l\neq y} z_l - z_y$

The Optimal Robust Classifier

Adversarial Accuracy of a DNN $\mathcal{C}$ wrt an attack radius ε: intrinsic robustness measurement.

$$\mathrm{AA}_{\mathcal{D}}(\mathcal{C},\varepsilon) = \mathbb{P}_{(x,y)\sim\mathcal{D}}\left(\forall \overline{x} \in \mathbb{B}(x,\varepsilon)\,(\widehat{\mathcal{C}}(\overline{x}) = y)\right)$$

Carlini-Wagner loss function: $\mathrm{Loss}_{\mathrm{cw}}(z,y) = \max_{l\in[m],l\neq y} z_l - z_y$

Theorem (Gao-Liu-Yu, 2022)

The game using loss function $\mathrm{Loss}_{\mathrm{cw}}$ has a Stackelberg equilibrium $(\Theta^*_{\mathrm{cw}}, A^*_{\mathrm{cw}})$, and $\mathcal{C}_{\Theta^*_{\mathrm{cw}}}$ is the optimal robust DNN against adversarial attacks:

$$\mathrm{AA}_{\mathcal{D}}(\mathcal{C}_{\Theta^*_{\mathrm{cw}}},\varepsilon) \geq \mathrm{AA}_{\mathcal{D}}(\mathcal{C}_{\Theta},\varepsilon),\ \forall\Theta \in [-E,E]^K$$

The Optimal Robust Classifier

Adversarial Accuracy of a DNN $\mathcal{C}$ wrt an attack radius ε: intrinsic robustness measurement.

$$\mathrm{AA}_{\mathcal{D}}(\mathcal{C},\varepsilon) = \mathbb{P}_{(x,y)\sim\mathcal{D}}\left(\forall \overline{x} \in \mathbb{B}(x,\varepsilon)\,(\widehat{\mathcal{C}}(\overline{x}) = y)\right)$$

Carlini-Wagner loss function: $\mathrm{Loss}_{\mathrm{cw}}(z,y) = \max_{l\in[m],l\neq y} z_l - z_y$

Theorem (Gao-Liu-Yu, 2022)

The game using loss function $\mathrm{Loss}_{\mathrm{cw}}$ has a Stackelberg equilibrium $(\Theta^*_{\mathrm{cw}}, A^*_{\mathrm{cw}})$, and $\mathcal{C}_{\Theta^*_{\mathrm{cw}}}$ is the optimal robust DNN against adversarial attacks:

$$\mathrm{AA}_{\mathcal{D}}(\mathcal{C}_{\Theta^*_{\mathrm{cw}}},\varepsilon) \geq \mathrm{AA}_{\mathcal{D}}(\mathcal{C}_{\Theta},\varepsilon),\ \forall\Theta \in [-E,E]^K$$

AT was recognized “the most successful empirical defense to date,”
“it is impossible to tell ... is truly robust.” (Cohen et at, 2019)
“it has shortages like ... non-provable.” (Bai et at, 2020)

Tradeoff Phenomenon: (CIFAR10)

DNN	ϵ	Normal		With AT	
		Accuracy	Adv. Accu	Accuracy	Adv. Accu
Resnet18	8/255	94%	0%	84%	52%
Resnet18	16/255	94%	0%	65%	35%
VGG16	8/255	93%	0%	79%	49%
VGG16	16/255	93%	0%	59%	31%

Tradeoff Between Robustness and Accuracy

Tradeoff Phenomenon: (CIFAR10)

DNN	ϵ	Normal		With AT	
		Accuracy	Adv. Accu	Accuracy	Adv. Accu
Resnet18	8/255	94%	0%	84%	52%
Resnet18	16/255	94%	0%	65%	35%
VGG16	8/255	93%	0%	79%	49%
VGG16	16/255	93%	0%	59%	31%

Tradeoff problem can be described as a bi-level optimization problem:

$$\Theta_o^* = \arg\min_{\Theta^*} \phi(\Theta^*)$$
$$\text{subject to } \Theta^* = \arg\min_{\Theta\in\mathcal{S}_c} \max_{A\in\mathcal{S}_a} \phi_{\rm cw}(\Theta, A)$$

For a DNN which is not a robust memorization for $\mathcal{D}$, tradeoff indeed happens.

Smale's 18th Problem: Limits of Intelligence

"This project requires the development of a mathematical model of intelligence, with variations to take into account the differences between kinds of intelligence."

Smale's 18th Problem: Limits of Intelligence

"This project requires the development of a mathematical model of intelligence, with variations to take into account the differences between kinds of intelligence."

- For a given data set $\mathcal{D}$, there exists a DNN which can correctly classify $\mathcal{D}$, but $\mathcal{D}$ is not computable. (Colbrook-Antun-Hansen, 21)

Smale's 18th Problem: Limits of Intelligence

"This project requires the development of a mathematical model of intelligence, with variations to take into account the differences between kinds of intelligence."

- For a given data set $\mathcal{D}$, there exists a DNN which can correctly classify $\mathcal{D}$, but $\mathcal{D}$ is not computable. (Colbrook-Antun-Hansen, 21)
- DNNs are Kolmogorov-optimal approximants for certain function classes. (Bölcskei, 21)

Smale's 18th Problem: Limits of Intelligence

"This project requires the development of a mathematical model of intelligence, with variations to take into account the differences between kinds of intelligence."

- For a given data set $\mathcal{D}$, there exists a DNN which can correctly classify $\mathcal{D}$, but $\mathcal{D}$ is not computable. (Colbrook-Antun-Hansen, 21)
- DNNs are Kolmogorov-optimal approximants for certain function classes. (Bölcskei, 21)
- Tradeoff between accuracy and robustness. If a DNN achieves optimal robustness, then its accuracy is confined.

Summary on the Optimal Robust DNNs

- Adversarial training with CW loss gives the optimal robust DNNs.
- But, the adversarial accuracy (CIFAR10) for the best DNN is still not high $60\% - 70\%$.
- Does there exist provable adversarially robust classifiers?

Information-theoretically Safe Bias-Classifier against Adversaries

Bias Classifier

$\mathcal{C}: \mathbb{I}^d \to \mathbb{R}^m$ a DNN with Relu as activation function

For any $x \in \mathbb{I}^d$, $\mathcal{C}(x) = W_x x + B_x = W_{\mathcal{C}}(x) + B_{\mathcal{C}}(x)$

where $W_x \in \mathbb{R}^{m \times d}$ and $B_x \in \mathbb{R}^m$

Bias Classifier: Piecewise constant

$$B_{\mathcal{C}}(x) = \mathcal{C}(x) - W_{\mathcal{C}}(x) = \mathcal{C}(x) - \frac{\nabla \mathcal{C}(x)}{\nabla x} \cdot x$$

Bias Classifier

$\mathcal{C} : \mathbb{I}^d \to \mathbb{R}^m$ a DNN with Relu as activation function

For any $x \in \mathbb{I}^d$, $\mathcal{C}(x) = W_x x + B_x = W_{\mathcal{C}}(x) + B_{\mathcal{C}}(x)$

where $W_x \in \mathbb{R}^{m\times d}$ and $B_x \in \mathbb{R}^m$

Bias Classifier: Piecewise constant

$$B_{\mathcal{C}}(x) = \mathcal{C}(x) - W_{\mathcal{C}}(x) = \mathcal{C}(x) - \frac{\nabla \mathcal{C}(x)}{\nabla x} \cdot x$$

Theorem (Existence of Bias Classifier)

For any data set and $\epsilon > 0$, there exists a DNN $\mathcal{C}$ such that $B_{\mathcal{C}}(x)$ gives the correct label with probability $> 1 - \epsilon$.

Training the Bias Classifier

Normal training:

$\min_{\Theta} \sum_{(x,y)\in\mathcal{S}} \mathrm{Loss}(\mathcal{C}(x), y)$

Adversarial training:

$\min_{\Theta} \max_{||\zeta||<\varepsilon} \sum_{(x,y)\in\mathcal{S}} \mathrm{Loss}(\mathcal{C}(x+\zeta), y)$

Adversarial training for Bias Classifier:

$\min_{\Theta} \max_{||\zeta||<\varepsilon} \sum_{(x,y)\in\mathcal{S}} [\mathrm{Loss}(B_{\mathcal{C}}(x+\zeta), y) + \gamma L_{\mathrm{ce}}(\mathcal{C}(x+\zeta), y)]$

Accuracies of Network Lenet-5 for MNIST

	$W_{\mathcal{C}}$	$B_{\mathcal{C}}$	$\mathcal{C}$
Normal training	98.80%	15.62%	99.09%
Adversarial training	90.61%	98.77%	99.19%
Bias Adversarial training	0.28%	99.09 %	99.43%

Information-theoretically Safety

Borrowed from cryptography, the ciphertext yields no information regarding the plaintext if the cyphers are perfectly random.

Information-theoretically Safety

Borrowed from cryptography, the ciphertext yields no information regarding the plaintext if the cyphers are perfectly random.

Original-model Gradient Based Attack for $B_{\mathcal{C}}$:

$\mathcal{A}(x, B_{\mathcal{C}}) = x + \rho\, \mathrm{sign}(\frac{\nabla \phi(\Theta, x)}{\nabla x}) = x + \rho\, \mathcal{D}_{\mathcal{A}}(x)$,
where $\mathcal{D}_{\mathcal{A}}(x) \in \{-1, 1\}^d$ is attacking direction

Information-theoretically Safety

Borrowed from cryptography, the ciphertext yields no information regarding the plaintext if the cyphers are perfectly random.

Original-model Gradient Based Attack for $B_{\mathcal{C}}$:

$$\mathcal{A}(x, B_{\mathcal{C}}) = x + \rho\,\mathrm{sign}\left(\frac{\nabla\phi(\Theta, x)}{\nabla x}\right) = x + \rho\,\mathcal{D}_{\mathcal{A}}(x),$$

where $\mathcal{D}_{\mathcal{A}}(x) \in \{-1, 1\}^d$ is attacking direction

The attack is called information-theoretically safe:

The attack direction $\mathcal{D}_{\mathcal{A}}(x)$ is a random vector in $\{-1, 1\}^d$.

Information-theoretically Safety

Borrowed from cryptography, the ciphertext yields no information regarding the plaintext if the cyphers are perfectly random.

Original-model Gradient Based Attack for $B_{\mathcal{C}}$:

$$\mathcal{A}(x, B_{\mathcal{C}}) = x + \rho\, \mathrm{sign}(\frac{\nabla\phi(\Theta,x)}{\nabla x}) = x + \rho\, \mathcal{D}_{\mathcal{A}}(x),$$

where $\mathcal{D}_{\mathcal{A}}(x) \in \{-1,1\}^d$ is attacking direction

The attack is called information-theoretically safe:

The attack direction $\mathcal{D}_{\mathcal{A}}(x)$ is a random vector in $\{-1,1\}^d$.

The adversary creation rate under attack $\mathcal{A}$ is the rate of random samples to be adversaries:

$$\mathcal{C}(\mathcal{C}) = \mathbb{P}_{x\sim\mathcal{D}, V\in\{-1,1\}^d}(\widehat{\mathcal{C}}(x+\rho V) \neq \widehat{\mathcal{C}}(x)),$$ which is quite small:

$\rho = 0.1$/MNIST/LeNet5	$\rho = 0.1$/CIFAR10/VGG19
$\mathcal{C}(\mathcal{C}) = 0.88\%$	$\mathcal{C}(\mathcal{C}) = 1.84\%$

Information-theoretically Safety Result (1)

FGSM Attack: $\mathcal{A}_1(\mathcal{C}, x) = x + \rho\,\mathrm{sign}(\frac{\nabla L(\mathcal{C}(x), y)}{\nabla x})$.

DNN: $\widetilde{\mathcal{C}}(x) = \mathcal{C}(x) + W_R \cdot x$, where W_R is a random matrix

Information-theoretically Safety Result (1)

FGSM Attack: $\mathcal{A}_1(\mathcal{C}, x) = x + \rho\,\mathrm{sign}(\frac{\nabla L(\mathcal{C}(x), y)}{\nabla x})$.

DNN: $\widetilde{\mathcal{C}}(x) = \mathcal{C}(x) + W_R \cdot x$, where W_R is a random matrix

Random matrix $\mathcal{M}_{m,n}(\lambda)$: i-th row in $\pm[(2i-1), 2i]\lambda$

Theorem (Binary Classification)

If $W_R \sim \mathcal{M}_{m,n}(\lambda)$ s.t. $|\mathbf{J}(\mathcal{C}(x))|_\infty < \lambda/2$,
then $B_{\widetilde{\mathcal{C}}}$ is information-theoretically safe against the attack $\mathcal{A}_1$.

Information-theoretically Safety Result (1)

FGSM Attack: $\mathcal{A}_1(\mathcal{C}, x) = x + \rho\, \mathrm{sign}(\frac{\nabla L(\mathcal{C}(x), y)}{\nabla x})$.

DNN: $\widetilde{\mathcal{C}}(x) = \mathcal{C}(x) + W_R \cdot x$, where W_R is a random matrix

Random matrix $\mathcal{M}_{m,n}(\lambda)$: i-th row in $\pm[(2i-1), 2i]\lambda$

Theorem (Binary Classification)

If $W_R \sim \mathcal{M}_{m,n}(\lambda)$ s.t. $|\mathbf{J}(\mathcal{C}(x))|_\infty < \lambda/2$,
then $B_{\widetilde{\mathcal{C}}}$ is information-theoretically safe against the attack $\mathcal{A}_1$.

Random matrix $\mathcal{U}_{m,n}(\lambda)$: entries $|u| \le \lambda$.

Theorem (Binary Classification)

If $W_R \sim \mathcal{U}_{m,n}(\lambda)$ s.t. $|\mathbf{J}(\mathcal{C}(x))|_\infty < \mu/2$ and $\lambda > n\mu/\ln(1+\epsilon)$, then
$\mathcal{C}(B_{\widetilde{\mathcal{C}}}, \mathcal{A}_1) \le (1+\epsilon)\mathcal{C}(\mathcal{C})$. (approximate information-theoretically safe)

Information-theoretically Safety Result (2)

Signed margin attack (Carlini-Wagner):

$$\mathcal{A}_2(x, \widetilde{\mathcal{C}}) = x + \rho\,\mathrm{sign}\left(\frac{\nabla \widetilde{\mathcal{C}}_{n_x}(x)}{\nabla x} - \frac{\nabla \widetilde{\mathcal{C}}_y(x)}{\nabla x}\right)$$

where y is the label of x, $n_x = \arg\max_{i \neq y}\{\mathcal{C}_i(x)\}$

DNN: $\widetilde{\mathcal{C}}(x) = \mathcal{C}(x) + W_R \cdot x$

Information-theoretically Safety Result (2)

Signed margin attack (Carlini-Wagner):

$$\mathcal{A}_2(x,\widetilde{\mathcal{C}}) = x + \rho\,\mathrm{sign}\left(\frac{\nabla\widetilde{\mathcal{C}}_{n_x}(x)}{\nabla x} - \frac{\nabla\widetilde{\mathcal{C}}_y(x)}{\nabla x}\right)$$

where y is the label of x, $n_x = \arg\max_{i\neq y}\{\mathcal{C}_i(x)\}$

DNN: $\widetilde{\mathcal{C}}(x) = \mathcal{C}(x) + W_R \cdot x$

Theorem

If $W_R \in \mathcal{M}_{m,n}(\lambda)$ s.t. $|\mathbf{J}(\mathcal{C}(x))|_\infty < \lambda/2$, then $B_{\widetilde{\mathcal{C}}}$ is information-theoretically safe against the attack $\mathcal{A}_2(\widetilde{\mathcal{C}})$.

If $W_R \sim \mathcal{U}_{m,n}(\lambda)$, $|\mathbf{J}(\mathcal{C}(x))|_\infty < \mu/2$, and $\lambda > \mu n/(\epsilon C(\mathcal{C},\rho))$, then $C(B_{\widetilde{\mathcal{C}}},\mathcal{A}_1) \leq (1+\epsilon)C(\mathcal{C})$. (approximate information-theoretically safe)

Experiments: Adversary Creation Rate

Attack	DNN	MNIST			CIFAR10		
		1-1	1-2	1-3	1-1	1-2	1-3
White-box	$B_{\mathcal{C}}$	2%	6%	22%	41%	58%	77%
	$\mathcal{C}$	3%	17%	55%	54%	77%	90%

Experiments: Adversary Creation Rate

Attack	DNN	MNIST			CIFAR10		
		1-1	1-2	1-3	1-1	1-2	1-3
White-box	$B_{\mathcal{C}}$	2%	6%	22%	41%	58%	77%
	$\mathcal{C}$	3%	17%	55%	54%	77%	90%
Black-box	$B_{\mathcal{C}}$	3%	6%	18%	27%	36%	43%
	$\mathcal{C}$	2%	3%	26%	30%	36%	48%

Experiments: Adversary Creation Rate

Attack	DNN	MNIST			CIFAR10		
		1-1	1-2	1-3	1-1	1-2	1-3
White-box	$B_{\mathcal{C}}$	2%	6%	22%	41%	58%	77%
	$\mathcal{C}$	3%	17%	55%	54%	77%	90%
Black-box	$B_{\mathcal{C}}$	3%	6%	18%	27%	36%	43%
	$\mathcal{C}$	2%	3%	26%	30%	36%	48%
ITS	$B_{\mathcal{C}}$	1%	2%	2%	19%	20%	22%
	$\mathcal{C}$	1%	2%	2%	19%	20%	21%
Accuracy	$B_{\mathcal{C}}$	99.12%			82.84%		
	$\mathcal{C}$	99.19%			81.23%		

Experiments: Adversary Creation Rate

Attack	DNN	MNIST			CIFAR10		
		1-1	1-2	1-3	1-1	1-2	1-3
White-box	$B_{\mathcal{C}}$	2%	6%	22%	41%	58%	77%
	$\mathcal{C}$	3%	17%	55%	54%	77%	90%
Black-box	$B_{\mathcal{C}}$	3%	6%	18%	27%	36%	43%
	$\mathcal{C}$	2%	3%	26%	30%	36%	48%
ITS	$B_{\mathcal{C}}$	1%	2%	2%	19%	20%	22%
	$\mathcal{C}$	1%	2%	2%	19%	20%	21%
Accuracy	$B_{\mathcal{C}}$	99.12%			82.84%		
	$\mathcal{C}$	99.19%			81.23%		

1. Bias classifier is more robust than DNNs with similar sizes.
2. Bias classifier can be made provably safe against the original-model gradient-based attack.

Summary

- **Robust Memorization**: There exist optimal robust DNNs with width $O(d)$ and depth $O(N)$.
- **Adversarial Stackelberg Game**: For DNNs with given width and depth, the equilibrium of the game gives optimal robust DNNs against adversarial attacks.
- **Bias Classifier**: Information-theoretically safe against original-model gradient-based attack.

Thanks to my students: Lijia Yu, Yihang Wang, Shuang Liu

Papers can be found: http://www.mmrc.iss.ac.cn/~xgao

Thanks!