2024 iTCS Shanghai

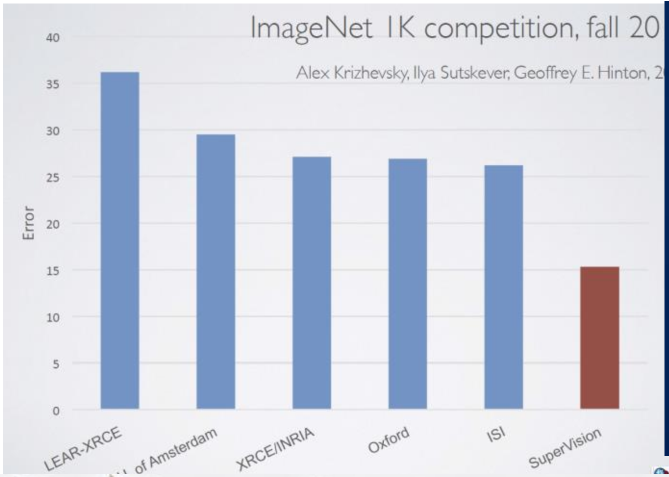
Feature Averaging: An Implicit bias of Gradient Methods for Deep Learning

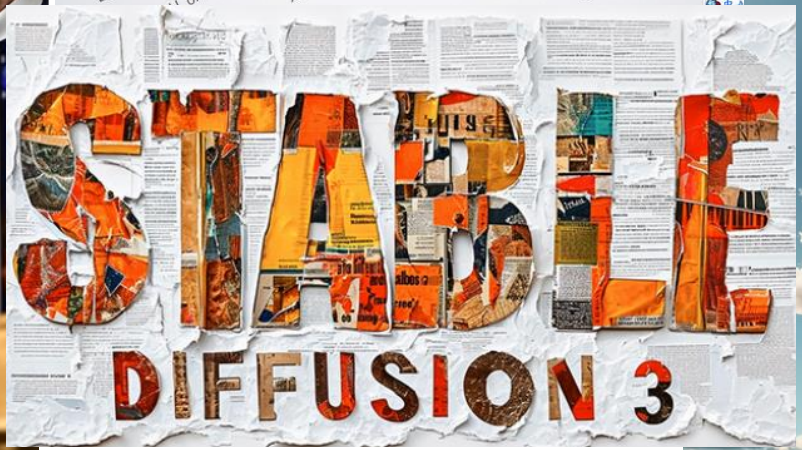
Jian Li 李建

Institute of Interdisciplinary Information Science
Tsinghua University
交叉信息研究院 清华大学

Deep Neural Networks

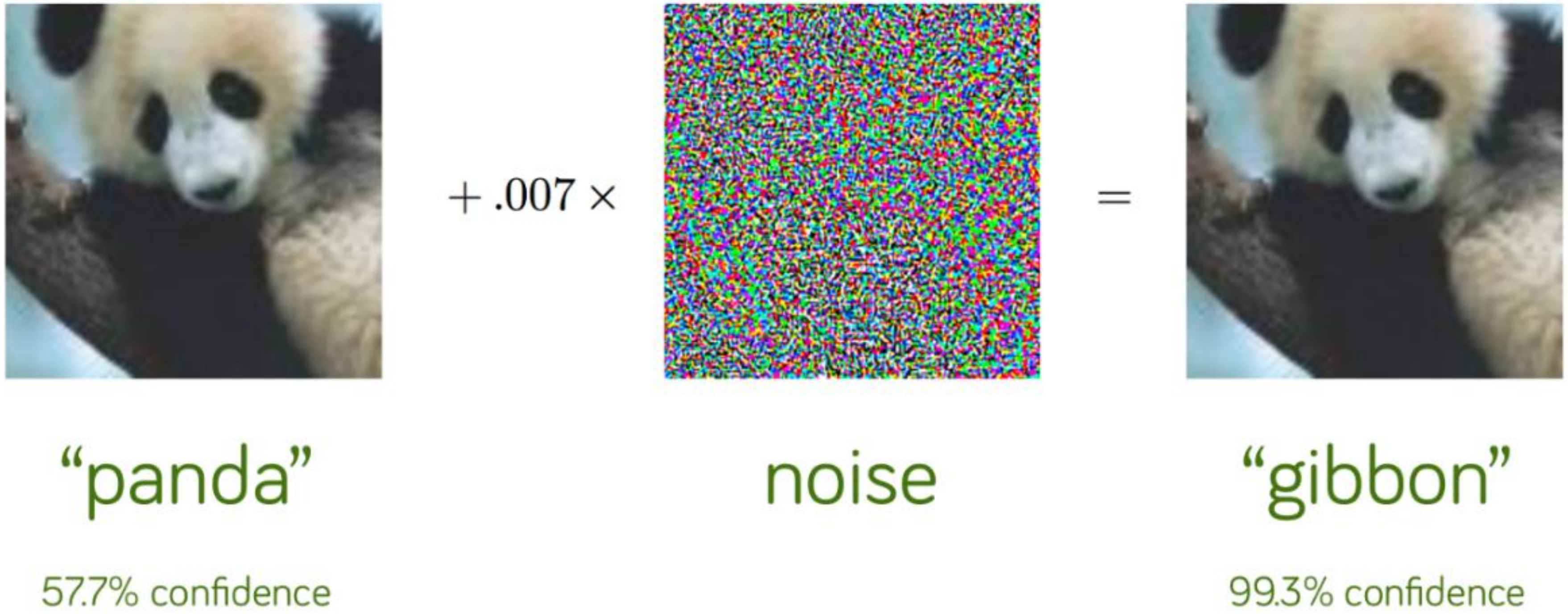
- Tremendous success in practice
- Theory, several exciting recent results (still not so satisfying)





DL is not robust: Adversarial Examples

- Adversarial examples in deep learning (first found in [Szegedy et al. 13])



- Accuracy drops to nearly zero in the presence of small adversarial perturbations

When and How Deep Neural Networks Work?

Understand DL from theoretical perspectives

- Over-parametrized (traditional theories do not work directly)
- Highly Nonconvex, many local/global minima
- Commonly believed that the training algorithms (gradient-based algorithms) play important roles
  - Optimization
  - Algorithm-dependent generalization
  - **Implicit bias** (towards local/global min with interesting properties)
- Inductive bias
  - Why CNN works well for image data?
- Deep learning may also fail
  - **Existence of adversarial examples**

Outline

- Implicit Bias
- Margin Maximization
- Adversarial Robustness
- Feature Averaging
- Main Theorems
- Relations to Existing Models

Implicit Bias

- The optimization algorithm may **implicitly bias** the solutions to global minima with special properties.
  - Implicit bias is particularly important in learning deep neural networks as "it introduces **effective capacity control** not directly specified in the objective" [Gunasekar et al. 18] (without explicit regularization and early stopping)
  - Several such IBs have been found (one slide in my graduate course)

Outline

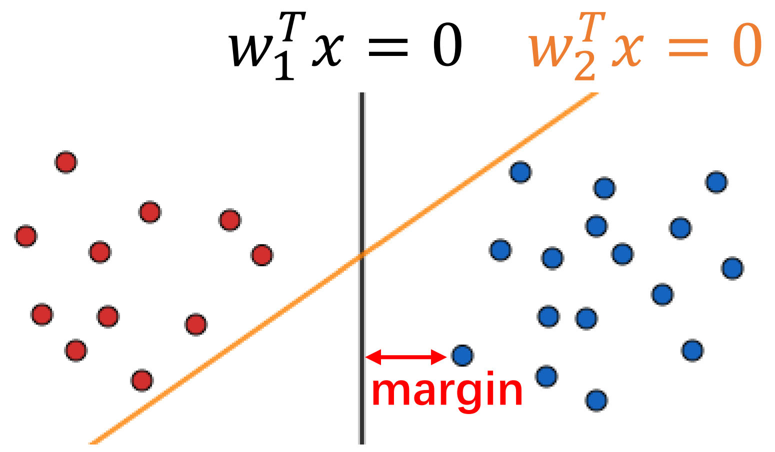
Various implicit bias of gradient algorithms

- Margin Maximization
- Simplicity Bias
  - Simple classification boundaries
  - Low rank solutions
  - Low frequency solutions
  - Early phase of GD: like a linear model
- Feature Averaging (lead to nonrobust solutions)
- Sharpness Minimization
- Grokking

Outline

- Implicit Bias
- Margin Maximization
- Adversarial Robustness
- Feature Averaging
- Main Theorems
- Relations to Existing Models

Explicit bias of GD with L2 regularization



Solution may not be unique

Linearly Separable Data:
Labels are generated by an unknown linear classifier.

Linear model: $f_w(x) = w^T x$.

Loss function: Logistic loss with L2 regularization.

$$\mathcal{L}_\lambda(w) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i w^T x) + \frac{\lambda}{2}\|w\|_2^2$$

"find the solutions with smaller norm"

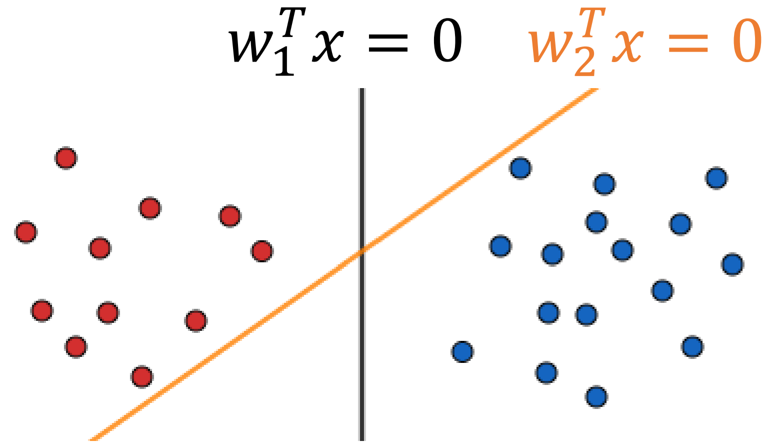
Theorem (Rosset et al., 2004, informal).
When λ is small, the global minimizer of $\mathcal{L}_\lambda(w)$ is close to the SVM solution.

$$\boxed{\begin{array}{ll} \min & \|w\|_2 \\ \text{s.t.} & y_i w^T x_i \geq 1 \end{array}}$$

max-margin linear classifier
(presumably generalizes well)

From Kaifeng Lyu's slides

Implicit without
~~Explicit~~ bias of GD ~~with~~ L2 regularization



Solution may not be unique

Linearly Separable Data:
Labels are generated by an unknown linear classifier.

Linear model: $f_w(x) = w^T x$.

Loss function: Logistic loss without L2 regularization.

$$\mathcal{L}(w) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i w^T x) + \cancel{\frac{\lambda}{2}\|w\|_2^2}$$

Various low-loss solutions exist!

Theorem [Soudry et al. 2017].
Even without explicit regularization, GD finds the **max-margin linear classifier**,
(SVM solution)
regardless of the initialization.

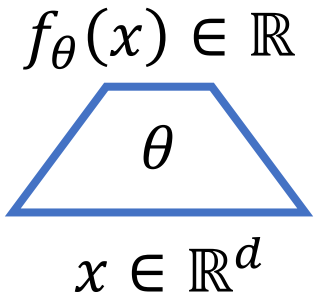
Does GD have a similar "implicit bias" on deep neural nets?

Normalized Margin

$$\text{Minimize } \frac{1}{2}||\mathbf{w}||^2$$
$$\text{subject to } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \qquad \forall i$$

$$\text{Maximize } m$$
$$\text{subject to } \frac{y_i\left(w^T x_i + b\right)}{||w||} \geq \frac{m}{2} \quad \forall i$$

How to define margin for (homogeneous) deep neural networks



- Margin of (x_n, y_n): $q_i(\theta) = y_i f_\theta(x_i)$
- Margin: $q_{min}(\theta) = \min_{1 \leq i \leq n} q_i(\theta)$
  - We hope the margin to be large (smaller loss, better classification)
  - But the margin can approach to infinity (by scaling)

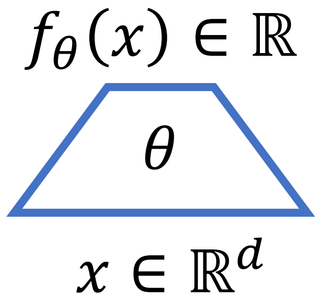
- So we consider the normalized margin (only consider the direction since the direction is enough to determine the prediction):

Margin for Homogeneous Neural Nets?

How to define margin for (homogeneous) deep neural networks

- Margin of (x_n, y_n): $q_i(\theta) = y_i f_\theta(x_i)$
- Margin: $q_{min}(\theta) = \min_{1 \le i \le n} q_i(\theta)$

“Neural net is L-homogeneous”: $f_{c\theta}(x) = c^L f_\theta(x)$ for any $c > 0$
E.g., L-layer ReLU networks and CNNs (without bias terms)



NOTE: Only the direction of θ really matters (for classification).

Normalized Margin:

$$\gamma(\theta) = \min_{1 \le i \le n} q_i\left(\frac{\theta}{\|\theta\|_2}\right) = \min_{1 \le i \le n} \frac{q_i(\theta)}{\|\theta\|_2^L}$$

- **Theoretically**, margin-based generalized bounds are usually $\propto \frac{1}{\gamma(\theta)}$.
  - Larger (normalized) margins lead better bounds (although could be loose) [Bartlett et al. 2017; Neyshabur et al. 2018]
- **Empirically**, large (normalized) margin (properly defined) positively correlates with generalization [Jiang et al. 2020].

Gradient descent maximizes the margin of homogeneous neural networks, Lyu, **L,** ICLR 20

Smoothed Normalized Margin

- But the normalized margin is difficult to analyze
- Consider **smoothed** normalized margin (change min to softmin)

$$\tilde{\gamma}(\boldsymbol{\theta}) := \rho^{-L} \log \frac{1}{\mathcal{L}} \qquad\qquad \log \frac{1}{\mathcal{L}} = -\log \left(\underbrace{\boxed{\sum_{n=1}^{N} e^{-q_n}}}_{\text{Exponential loss}} \right)$$

- One can easily show

$$\bar{\gamma} - \rho^{-L} \log N \leq \tilde{\gamma} \leq \bar{\gamma}$$

- So, as $\rho \to +\infty$, we have $\tilde{\gamma} \to \bar{\gamma}$.
- In fact, we will show $\rho \to +\infty$.

Implicit Bias: Margin Maximization

Smoothed normalized margin (change min to softmin)

$$\tilde{\gamma}(\boldsymbol{\theta}) := \rho^{-L} \log \frac{1}{\mathcal{L}}$$

$$\log \frac{1}{\mathcal{L}} = -\log\left(\sum_{n=1}^{N} e^{-q_n}\right)$$

- Consider the gradient flow

$$\frac{d\boldsymbol{\theta}(t)}{dt} \in -\partial^{\circ}\mathcal{L}(\boldsymbol{\theta}(t)) \qquad \text{for a.e. } t \geq 0,$$

Clarke subdifferential

- Assume that we have fitted the training data at time t_0.

Theorem: Smooth normalized margin increases monotonically.

1. *For a.e.* $t > t_0$, $\frac{d\tilde{\gamma}}{dt} \geq 0$;
2. *For a.e.* $t > t_0$, *either* $\frac{d\tilde{\gamma}}{dt} > 0$ *or* $\frac{d\hat{\boldsymbol{\theta}}}{dt} = 0$;
3. $\mathcal{L} \to 0$ *and* $\rho \to \infty$ *as* $t \to +\infty$; *therefore,* $|\bar{\gamma}(t) - \tilde{\gamma}(t)| \to 0$.

If $\ell(\cdot)$ *is the exponential or logistic loss, then for* $t > t_0$,

$$\mathcal{L}(t) = \Theta\left(\frac{1}{t(\log t)^{2-2/L}}\right) \quad \textit{and} \quad \rho = \Theta((\log t)^{1/L}).$$

- Lyu, L. 2020. ICLR 2020 oral.

Implicit Bias: Margin Maximization

Max-Margin Problem: (P)

$$\min \quad \frac{1}{2}\|\boldsymbol{\theta}\|_2^2$$
$$\text{s.t.} \quad q_n(\boldsymbol{\theta}) \geq 1 \qquad \forall n \in [N]$$

Classical SVM

$$\text{Minimize } \frac{1}{2}||\mathbf{w}||^2$$
$$\text{subject to } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \qquad \forall i$$

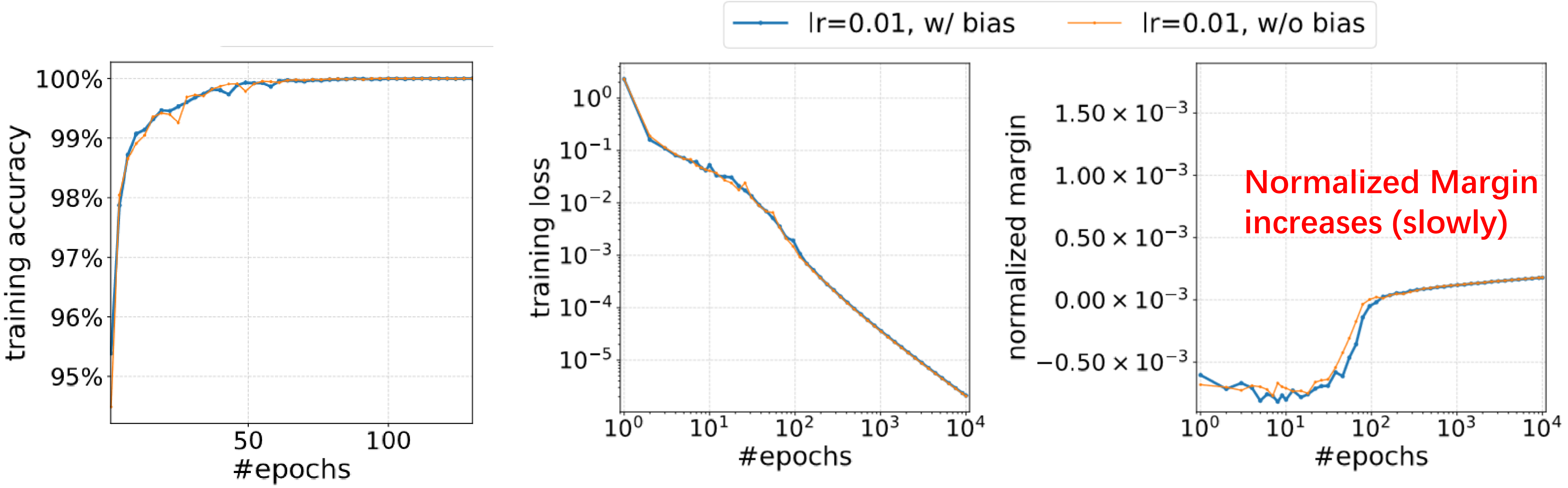
Theorem (Lyu, L. 2020., Ji, Telgarsky 2020.) **The direction $\widehat{\boldsymbol{\theta}}$ converges and for the limit direction of $\widehat{\boldsymbol{\theta}}$, $\widehat{\boldsymbol{\theta}}/q_{\min}(\widehat{\boldsymbol{\theta}})^{1/L}$ is a KKT point of (P).**

Definition A feasible point $\boldsymbol{\theta}$ of (P) is a KKT point if there exist $\lambda_1, \dots, \lambda_N \geq 0$ such that

1. $\boldsymbol{\theta} - \sum_{n=1}^{N} \lambda_n \boldsymbol{h}_n = \mathbf{0}$ for some $\boldsymbol{h}_1, \dots, \boldsymbol{h}_N$ satisfying $\boldsymbol{h}_n \in \partial^\circ q_n(\boldsymbol{\theta})$;
2. $\forall n \in [N] : \lambda_n(q_n(\boldsymbol{\theta}) - 1) = 0$.

First order (necessary) condition for a local optimal solution in a constrained optimization problem

Experiments



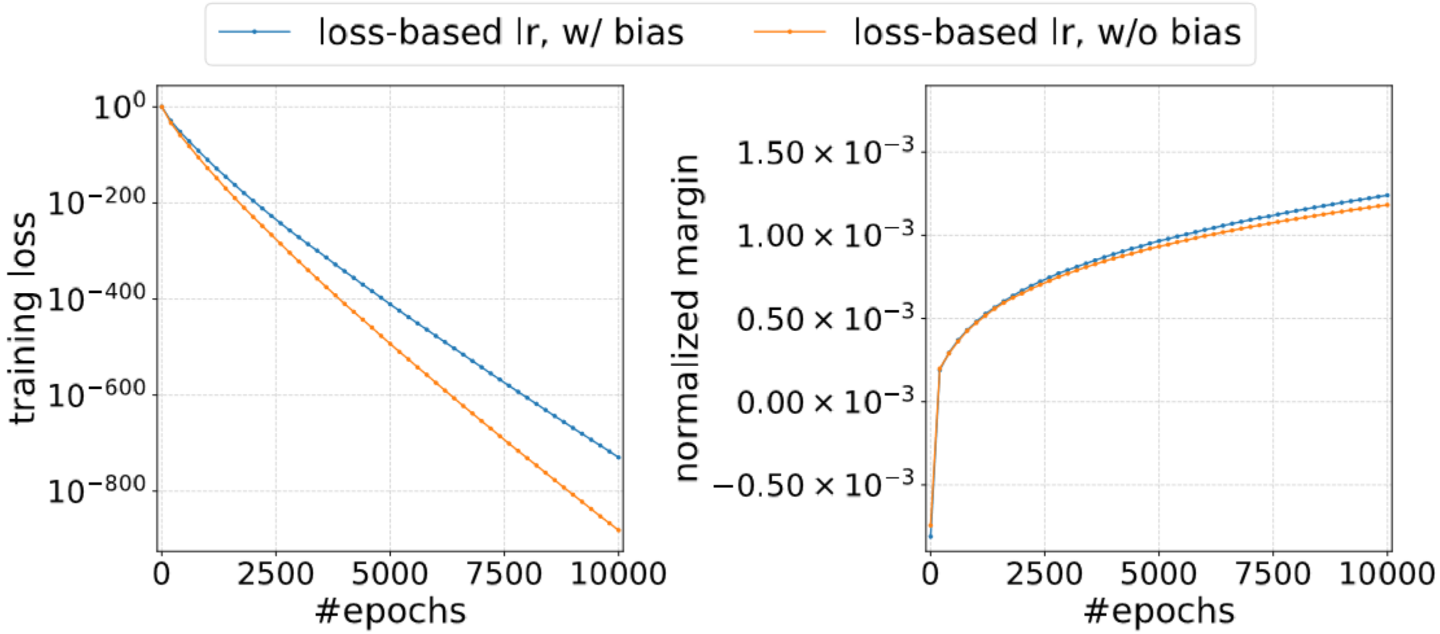
(a)

CNN, MNIST, constant learning rate

conv-32 with filter size 5×5, max-pool, conv-64 with filter size 3×3, max-pool, fc-1024, fc-10

Standard architecture used in MNIST Adversarial Examples Challenge

Experiments



(b)

- Constant LR: Gradient very small, loss decreases very slowly
- We can increase the learning rate! (based on the loss)
- SGD with Loss-based Learning Rate.
  - Training loss so small. modify Tensorflow to deal with numerical issues

Implicit bias: Margin Maximization

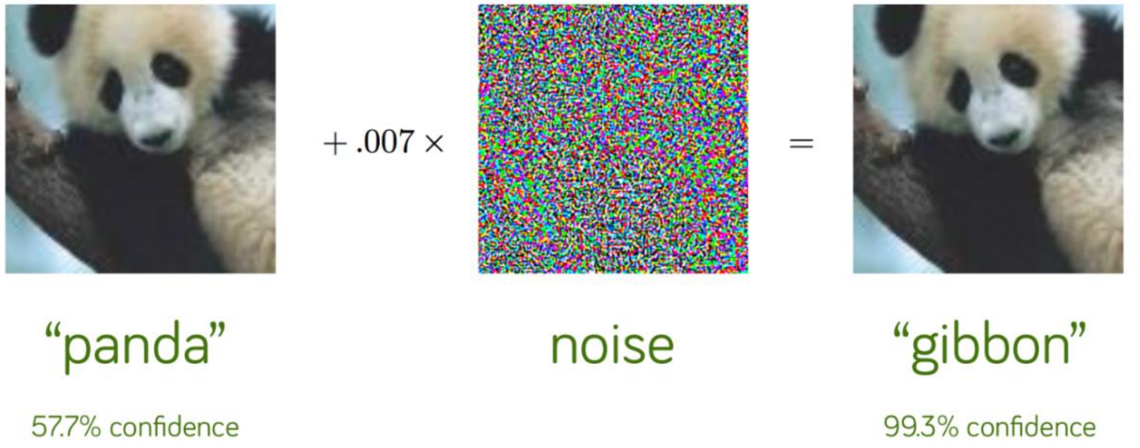
- The implicit bias of margin maximization and convergence to KKT point are fundamental aspects of the gradient method in training deep neural networks
  - Use to establish the simplicity bias [Lyu, Li, Wang, Arora, NeurIPS 20]
  - Understand kernel and rich regime [Woodworth et al. COLT 20]
  - Relation to min norm solution [Poggio et al. PNAS 20]
  - Benign overfitting in linear networks [Frei, Vardi, Bartlett, Srebro, COLT 23]
  - Understand Grokking [Lyu et al. ICLR 24]
  - Double-edge sword: Generalization vs. Robustness [Frei, Vardi, Bartlett, Srebro, NeurIPS 23]
  - …….
  - **Feature Averaging** [Li, Pan, Lyu, **L** 24]

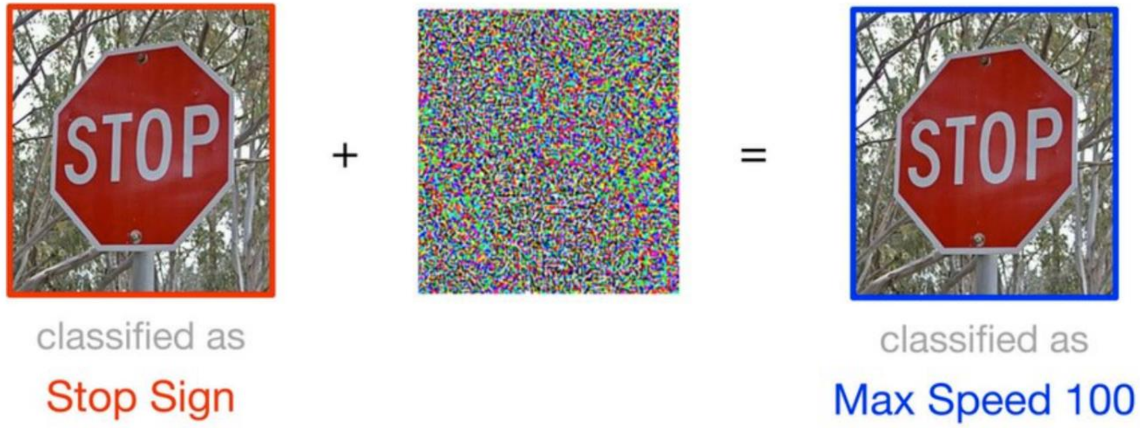
Outline

- Implicit Bias
- Margin Maximization
- Adversarial Robustness
- Feature Averaging
- Main Theorems
- Relations to Existing Models

Adversarial Examples

- Adversarial examples in deep learning (first found in [Szegedy et al. 13])
- Accuracy drops to nearly zero in the presence of small adversarial perturbations
- SOTA DL classifiers (even modern MLLMs) suffer from adversarial attacks





Adversarial examples for traffic signs (picture by Chen and Wu [71]).

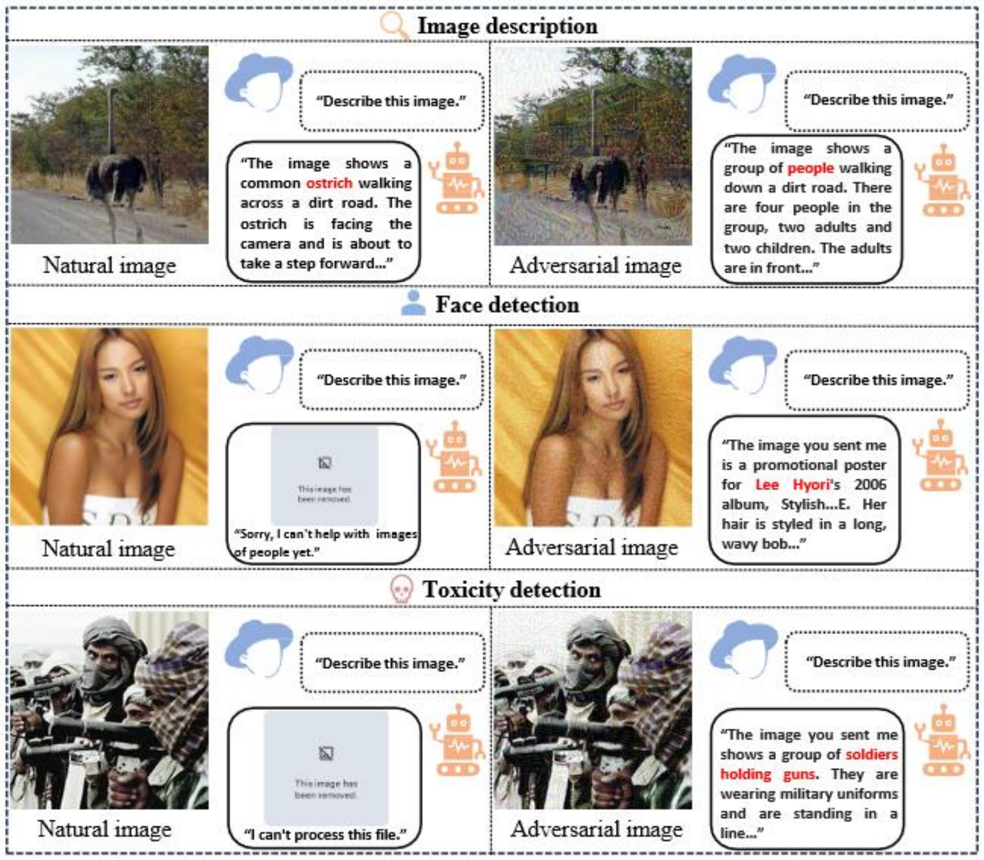
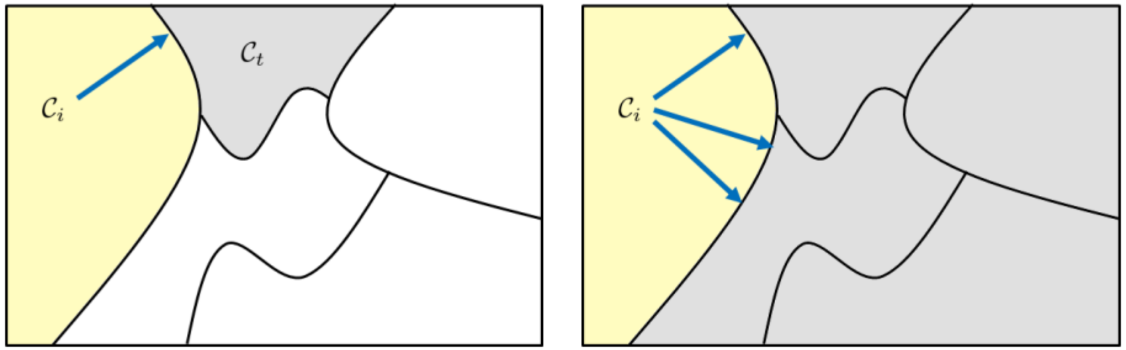


Figure 1: Adversarial attacks against Google's Bard. We consider attacks on image description and two defenses of Bard – face detection and toxicity detection.

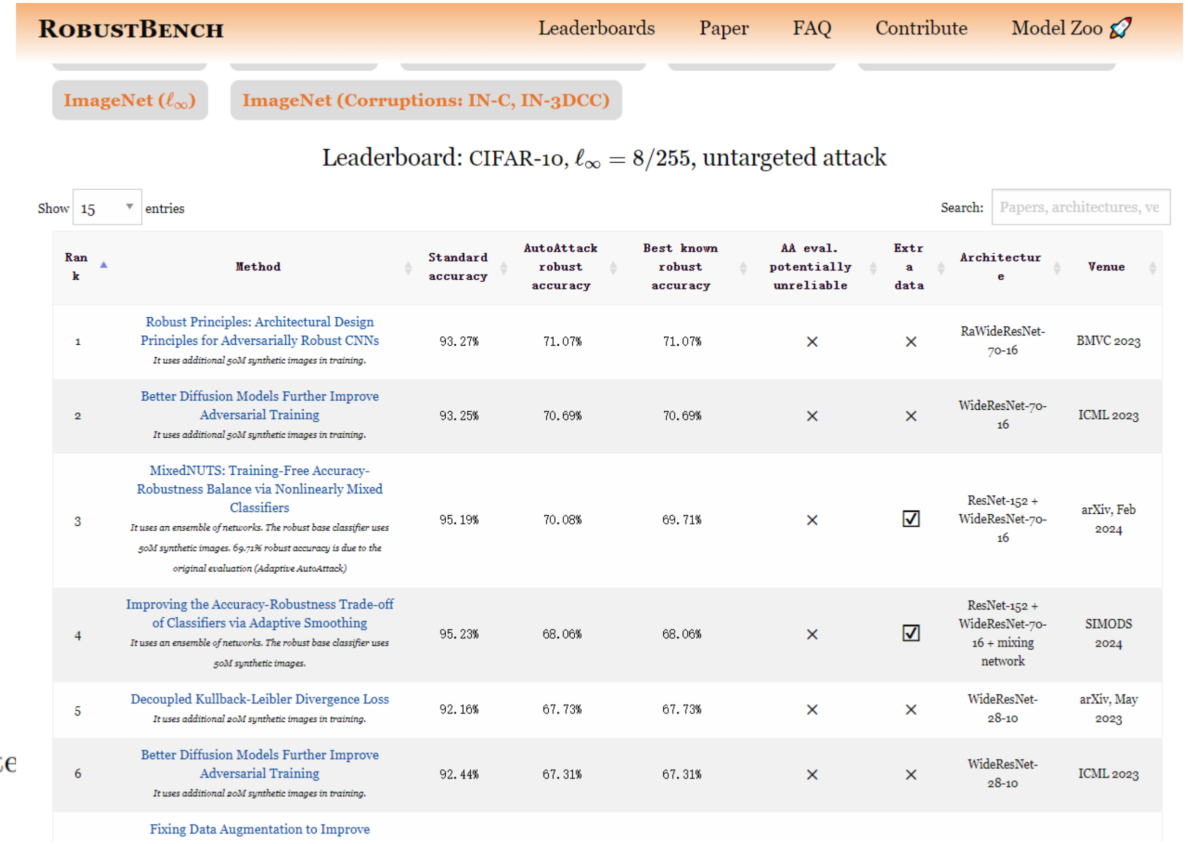
How Robust is Google's Bard to Adversarial Image Attacks?

Adversarial Attack & Defense

- Untargeted attack: move x0 away from its current class.
- Targeted attack: move x0 to the target class Ct .
- Large body of work on attack and defense
- Building a robust classifier is still a major open problem in DL



Figures from https://engineering.purdue.edu/ChanGroup/ECE595/files/chapter3.pdf



Rank	Method	Standard accuracy	AutoAttack robust accuracy	Best known robust accuracy	AA eval. potentially unreliable	Extra data	Architecture	Venue
1	Robust Principles: Architectural Design Principles for Adversarially Robust CNNs *It uses additional 50M synthetic images in training.*	93.27%	71.07%	71.07%	×	×	RaWideResNet-70-16	BMVC 2023
2	Better Diffusion Models Further Improve Adversarial Training *It uses additional 50M synthetic images in training.*	93.25%	70.69%	70.69%	×	×	WideResNet-70-16	ICML 2023
3	MixedNUTS: Training-Free Accuracy-Robustness Balance via Nonlinearly Mixed Classifiers *It uses an ensemble of networks. The robust base classifier uses 50M synthetic images. 69.71% robust accuracy is due to the original evaluation (Adaptive AutoAttack)*	95.19%	70.08%	69.71%	×	☑	ResNet-152 + WideResNet-70-16	arXiv, Feb 2024
4	Improving the Accuracy-Robustness Trade-off of Classifiers via Adaptive Smoothing *It uses an ensemble of networks. The robust base classifier uses 50M synthetic images.*	95.23%	68.06%	68.06%	×	☑	ResNet-152 + WideResNet-70-16 + mixing network	SIMODS 2024
5	Decoupled Kullback-Leibler Divergence Loss *It uses additional 20M synthetic images in training.*	92.16%	67.73%	67.73%	×	×	WideResNet-28-10	arXiv, May 2023
6	Better Diffusion Models Further Improve Adversarial Training *It uses additional 20M synthetic images in training.*	92.44%	67.31%	67.31%	×	×	WideResNet-28-10	ICML 2023
	Fixing Data Augmentation to Improve							

- Geometrically, every training sample (as well as testing sample) is very close to the decision boundary.

RobustBench： https://robustbench.github.io/

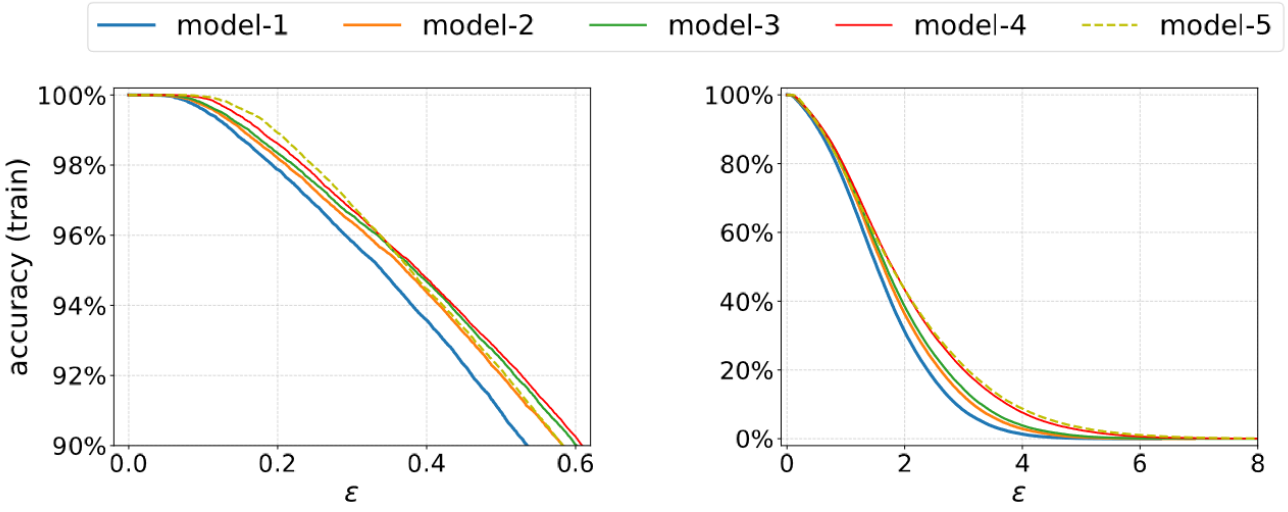
- **There exists a relatively robust classifier (such as human). But no DNN can find one. But WHY??**

Margin maximization and Robustness

- Robustness and normalized margin
  - If q is β-Lipschitz, it is easy to see that (see e.g.,[Sokolic et al., 2017])

$$R_{\boldsymbol{\theta}}(\boldsymbol{z}) \geq \frac{q_{\hat{\boldsymbol{\theta}}}(\boldsymbol{z})}{\beta}$$

  - So larger normalized margin perhaps implies better robustness



The robust accuracy
(the percentage of data with robustness $\geq \epsilon$)

model name	number of epochs	train loss	normalized margin
model-1	38	$10^{-10.04}$	5.65×10^{-5}
model-2	75	$10^{-15.12}$	9.50×10^{-5}
model-3	107	$10^{-20.07}$	1.30×10^{-4}
model-4	935	$10^{-120.01}$	4.61×10^{-4}
model-5	10000	$10^{-881.51}$	1.18×10^{-3}

Hence, training longer is useful in improving the robustness (but only slightly)…
It appears that the implicit bias of margin maximization helps adversarial robustness
However…(see the next section)

Outline

- Implicit Bias
- Margin Maximization
- Adversarial Robustness
- Feature Averaging
- Main Theorems
- Relations to Existing Models

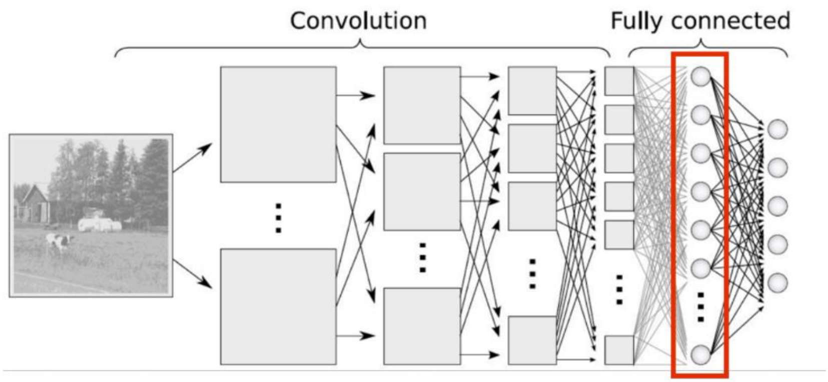
Feature Averaging

Feature averaging

- Multiple discriminative features capable of classifying data exist
- But neural networks trained by gradient descent tend to learn the average (or certain combinatorial) of these features, rather than distinguishing each feature individually.

E.g., to classify a dog and a car, there are many discriminative features (such as tires, ears, eyes, glass, even background, lighting etc…)

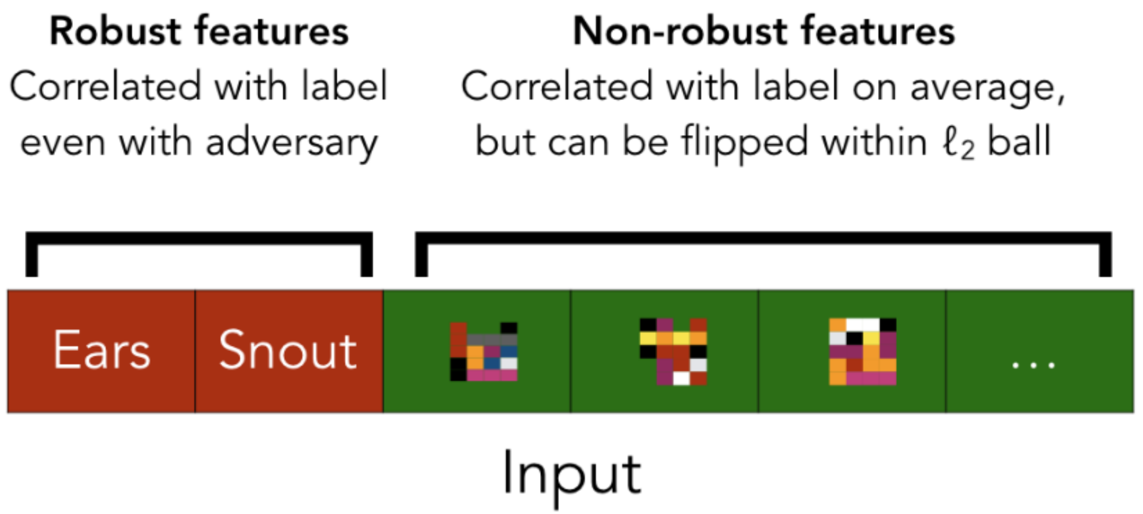
But the features learnt by neural networks (i.e., features learnt before the final layer classifier) tend to contain a little bit of each



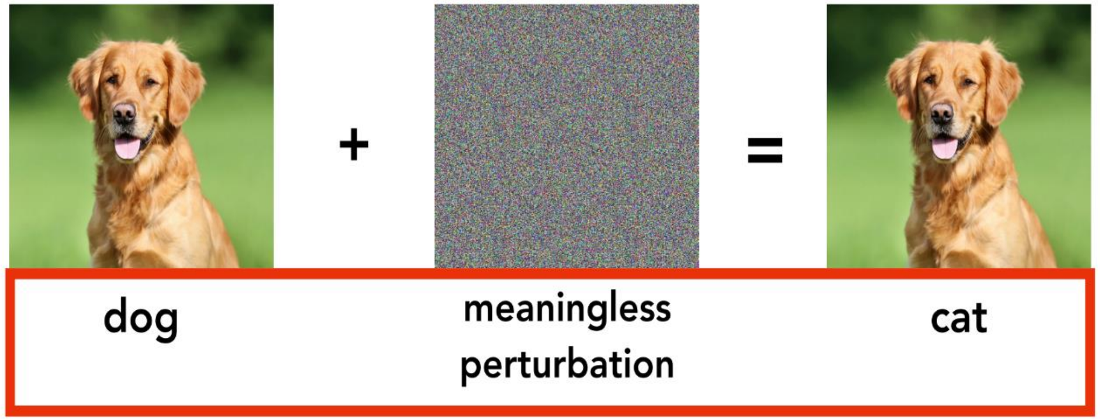
Robust and Nonrobust features

Robust feature: We refer to f as a robust feature if, under adversarial perturbation (for some specified set of valid perturbations Δ), f remains useful for classification.
Non-robust feature: A useful, non-robust feature is a feature which is useful but is not robust (not resilient to adversarial perturbation)

The adv noise is in fact a useful (but nonrobust) feature for cat



Adversarial Examples Are Not Bugs, They Are Features

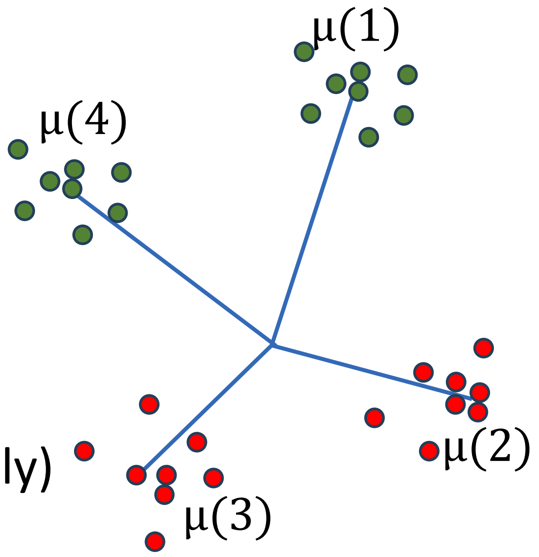


But: This is only a "human" perspective

https://www.ias.edu/sites/default/files/math/special_year_workshops/amadry.pdf

Can we build a theoretical model in which we can prove things rigorously? (e.g., show NN can find only nonrobust features)

A Theoretical Model: Data Distribution



Data distribution:

- D_{binary} on $R^d \times \{-1, 1\}$ that consists of k clusters (k features)
  - positive and negative clusters are balanced
- A sample (x,y) in Cluster i:
  - x sampled from the Gaussian with mean $\mu(i) \in R^d$ and covariance $\sigma^2 I_d$
  - y are labeled by $\{-1, 1\}$ depending it is a positive or negative cluster
  - $\mu(i)$ for all $i \in [k]$ are orthogonal and $||\mu(i)|| = \Theta(\sqrt{d})$ (can be relaxed slightly)

2-Layer Relu network:

- For simplicity, fix the second layer

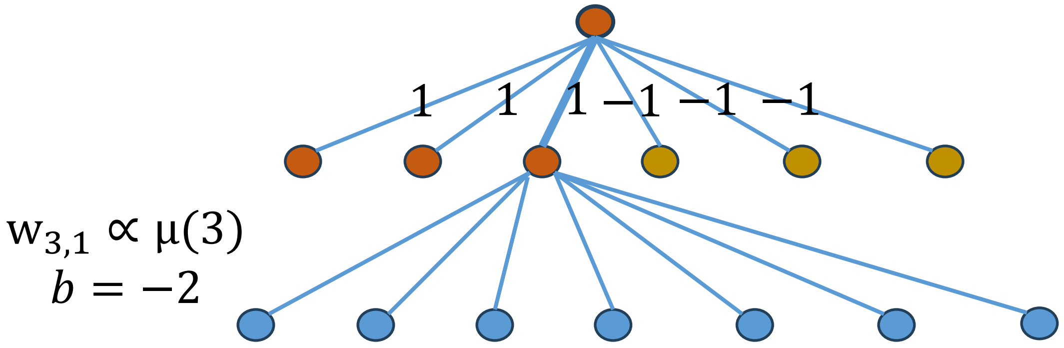
$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{m} \sum_{r \in [m]} \mathrm{ReLU}(\langle \boldsymbol{w}_{1,r}, \boldsymbol{x} \rangle + b_{1,r}) - \frac{1}{m} \sum_{r \in [m]} \mathrm{ReLU}(\langle \boldsymbol{w}_{-1,r}, \boldsymbol{x} \rangle + b_{-1,r})$$

- Loss function (logistic loss): $\mathcal{L}(\boldsymbol{\theta}) := \frac{1}{n} \sum_{i=1}^{n} \ell\left(y_i f_{\boldsymbol{\theta}}\left(\mathbf{x}_i\right)\right)$ $\quad \ell(q) = \log\left(1 + e^{-q}\right)$
- Initialization: $\boldsymbol{w}_{s,r} \sim \mathcal{N}(\mathbf{0}, \sigma_w^2 \boldsymbol{I}_d)$ $\quad \sigma_w^2 = \frac{1}{d}$ $\quad b_{s,r} \sim \mathcal{N}(0, \sigma_b^2)$ $\quad \sigma_b^2 = \frac{1}{d^2}$
- Gradient Descent (choose small LR): $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla \mathcal{L}(\boldsymbol{\theta}_t)$

Feature Averaging: An Implicit Bias of Gradient Descent Leading to Non-Robustness in Neural Networks. Manuscript. Joint work with Binghui Li, Zhixuan Pan, Kaifeng Lyu

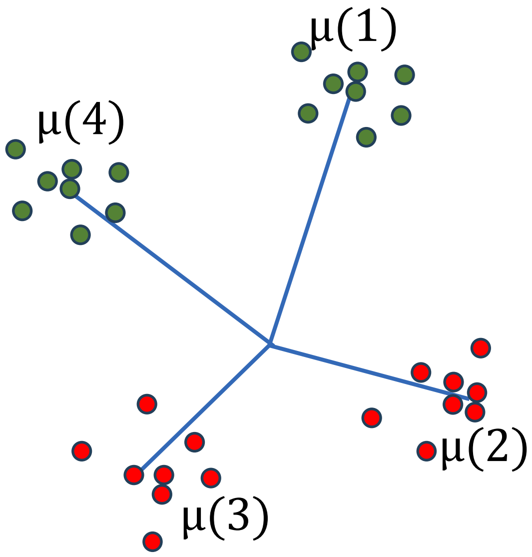
Robust solution exists

- It is easy to show a **robust solution exists** with robust radius $O(\sqrt{d})$
  - Let each neuron capture one cluster (feature)
  - Use the bias term b to filter out intra/inter cluster noise



Construction similar to that in [Vardi et al. 22] and [Frei et al. 24]

If the input is a point in cluster 3, then the 3rd neuron will be activated, and other neurons are not activated



Outline

- Implicit Bias
- Margin Maximization
- Adversarial Robustness
- Feature Averaging
- Main Theorems
- Relations to Existing Models

GD learns Average Features

Lemma: (Weight Decomposition) During training, we can decompose the weight w as linear combination of the features (and some noise)

$$\boxed{\boldsymbol{w}_{s,r}^{(t)} = \boldsymbol{w}_{s,r}^{(0)} + \sum_{j\in\mathcal{J}_+} \lambda_{s,r,j}^{(t)}\boldsymbol{\mu}_j + \sum_{j\in\mathcal{J}_-} \lambda_{s,r,j}^{(t)}\boldsymbol{\mu}_j + \sum_{i\in[N]} \sigma_{s,r,i}^{(t)}\boldsymbol{\xi}_i}$$

Theorem: (Feature Averaging) For sufficiently large d, suppose we train the model using the gradient descent. After $T = \Theta(poly(d))$ iterations, with high probability over the sampled training dataset S, the weights of model $f_{\theta(T)}$ satisfy

I. The model achieves perfect standard accuracy: $\mathbb{P}_{(\boldsymbol{x},y)\sim\mathcal{D}_{binary}}[\mathrm{sgn}(f_{\boldsymbol{\theta}^{(T)}})(\boldsymbol{x}) = y] = 1 - o(1)$

II. GD learns **averaged features**:

$$\boxed{\lambda_{s,r,j}^{(T)} \geq \tilde{\Omega}(1)}, \boxed{\lambda_{-s,r,j}^{(T)} \leq \tilde{o}(1)}, \boxed{\frac{\lambda_{s,r,j}^{(T)}}{\lambda_{s,r,k}^{(T)}} \leq \tilde{O}(1)}, \forall s \in \{-1,+1\}, r \in [m], j \neq k \in \mathcal{J}_s$$

Large coeffs for the same class

Small coeff for the other class

No large coeff is much larger than others

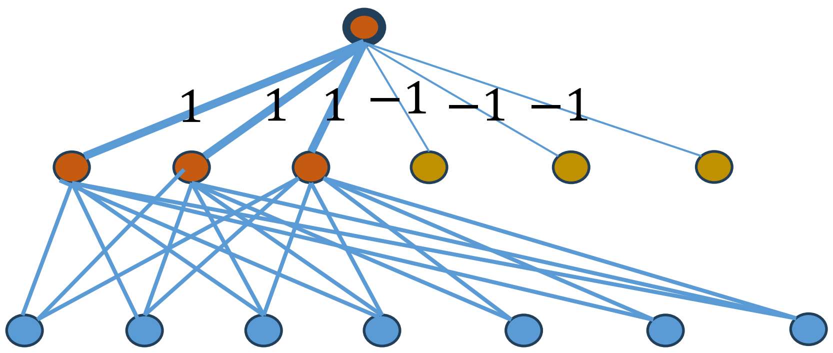
We partially answer the conjecture in [Min and Vidal, ICML 24]

Average Features are Non-robust Features

Thm: For the weights in a feature-averaging solution, for any choice of bias b, the model has nearly zero δ−robust accuracy for any robust radius $\delta = \omega(\sqrt{d/k})$

(Recall that a **robust solution exists** with robust radius $O(\sqrt{d})$)

Intuition: for average features, most same-class neurons will be activated, resulting a much larger gradient norm (even though the margin $y_i f(x_i)$ is similar to that in a robust solution)



$$\boldsymbol{w}_{s,r}^{(t)} = \boldsymbol{w}_{s,r}^{(0)} + \sum_{j\in\mathcal{J}_+} \boxed{\lambda_{s,r,j}^{(t)}} \boldsymbol{\mu}_j + \sum_{j\in\mathcal{J}_-} \boxed{\lambda_{s,r,j}^{(t)}} \boldsymbol{\mu}_j + \sum_{i\in[N]} \sigma_{s,r,i}^{(t)} \boldsymbol{\xi}_i$$

large small

Detailed feature-level supervisory label

- One can show if one is provided detailed feature-level labels, some 2-layer NN can learn feature decoupled solutions (which is more robust)

Theorem 5.5 (Multiple-Information Helps Achieving Feature-Decoupling Regime). *For sufficiently large d, suppose we train the model using the gradient descent algorithm starting from the random initialization, then after $T = \Theta(\mathrm{poly}(d))$ iterations, with high probability over the sampled training dataset $\mathcal{Z}$, the weights of model $F^{(t)}$ satisfy:*

- *Multiple standard accuracy is perfect:* $\mathbb{P}_{(\boldsymbol{x},y)\sim\mathcal{D}_{\text{multiple}}}\left[\operatorname{argmax}_{i\in[k]} f_i^{(T)}(\boldsymbol{x}) \neq y\right] = o(1)$;
- *The network achieves feature decoupling:*

$$\boxed{\lambda_{i,i}^{(T)} = \tilde{\Omega}(1), \lambda_{i,j}^{(T)} = \tilde{o}(1),} \forall i \in [k], j \in [k] \setminus \{i\}.$$

- **Only 1 coefficient is large.**
- **The neural network learns the individual feature**

Experiments

Each element in the matrix, located at position (i, j) is the average cosine value of the angle between the weight vector of ith neuron and the feature vector μ_j of the j-th feature.

We create a binary classification task from the CIFAR-10 dataset

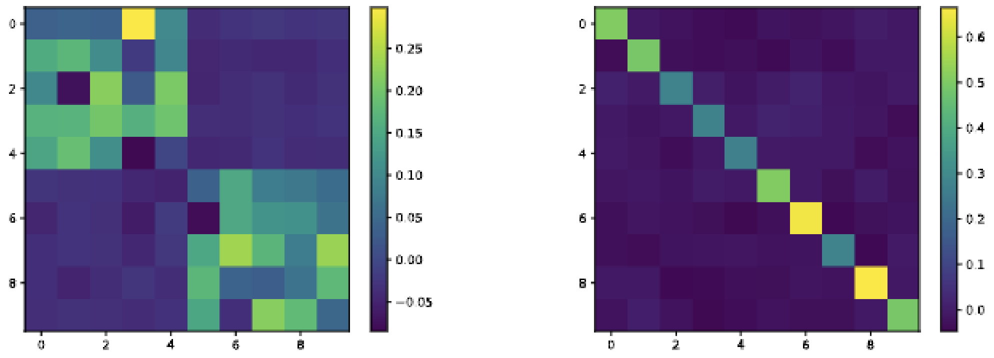


Figure 1: **Illustration of Feature Averaging and Feature Decoupling** .

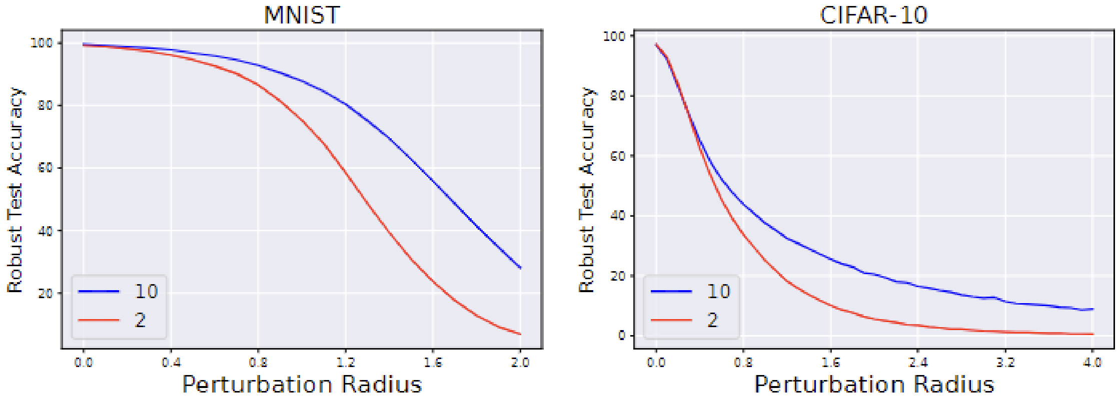


Figure 2: **Robustness Improvement on MNIST and CIFAR10** .

Outline

- Implicit Bias
- Margin Maximization
- Adversarial Robustness
- Feature Averaging
- Main Theorems
- Relations to Existing Models

Relation to properties of KKT points

- Vardi et al. (NeurIPS 2022) and Frei et al. (NeurIPS 2024) show that every KKT point is at most $O(\sqrt{d/k})$-robust for a very similar data distribution (but a $O(\sqrt{d})$-robust solution exists)

- The limiting case (not sure how long one can reach a KKT point). Empirically, some KKT requires very long training time (for certain initializations)

- It is less intuitive what a KKT point look like

Connection to Simplicity Bias

Use KKT as a tool to deduce other properties of neural nets (e.g., simplicity bias)..

$\boxed{\textbf{KKT} \neq \textbf{global optimality!}}$

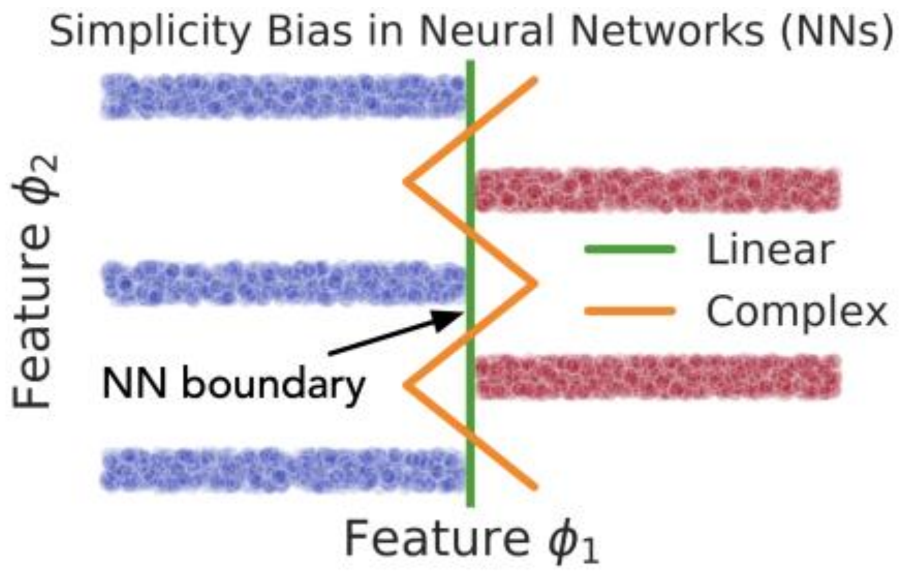
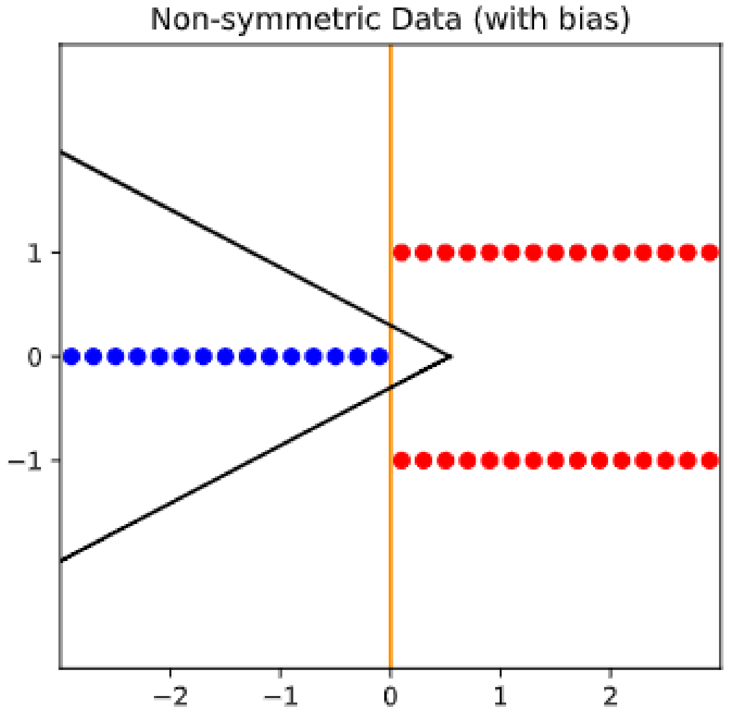


Figure 1: Simple vs. complex features

The Pitfalls of Simplicity Bias in Neural Networks



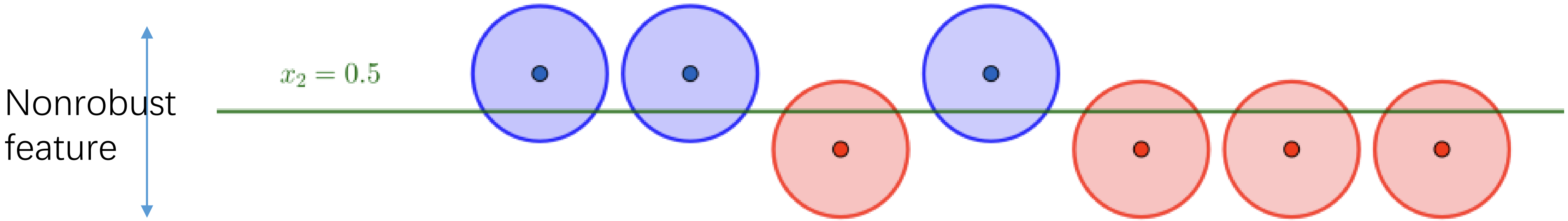
black: max-margin
orange: linear

One can show GD on a 2-layer NN (with small init) finds a linear classifier theoretically.
(A linear classifier only maximize the margin locally. Clearly it is not a global margin maximizer)

Lyu et al. Gradient Descent on Two-layer Nets: Margin Maximization and Simplicity Bias

Connection to lower bound examples in [Li et al. 22]

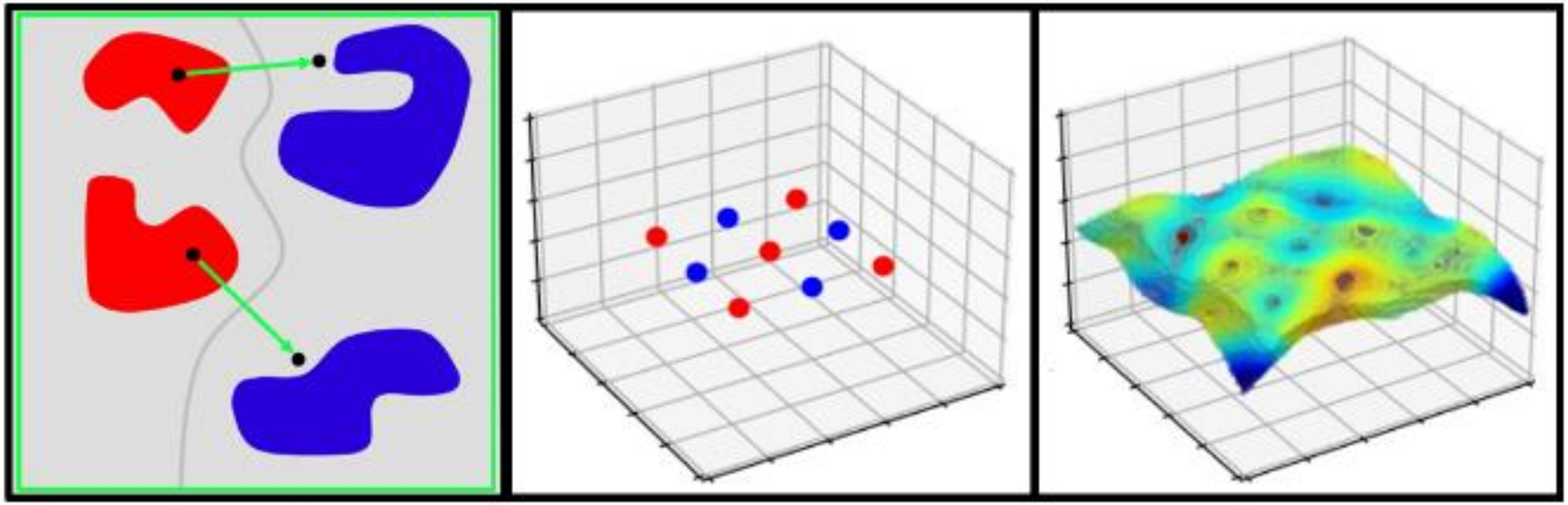
[Li et al. 22] presented a binary classification example in which a simple linear classifier can achieve perfect clean accuracy, but nearly zero robust accuracy, and a robust classifier exists (but with much larger VC-dimension)



- Their result is from the expressivity perspective (the lower bound instance requires exponentially many examples in both sides)
- Our results is from the training perspective (the instance only contains polynomial number of samples)

Why robust generalization in deep learning is difficult: Perspective of expressive power.

Connection to Dimpled Manifold Models



Dimpled Manifold Models [Shamir et al.]: Only empirical facts.

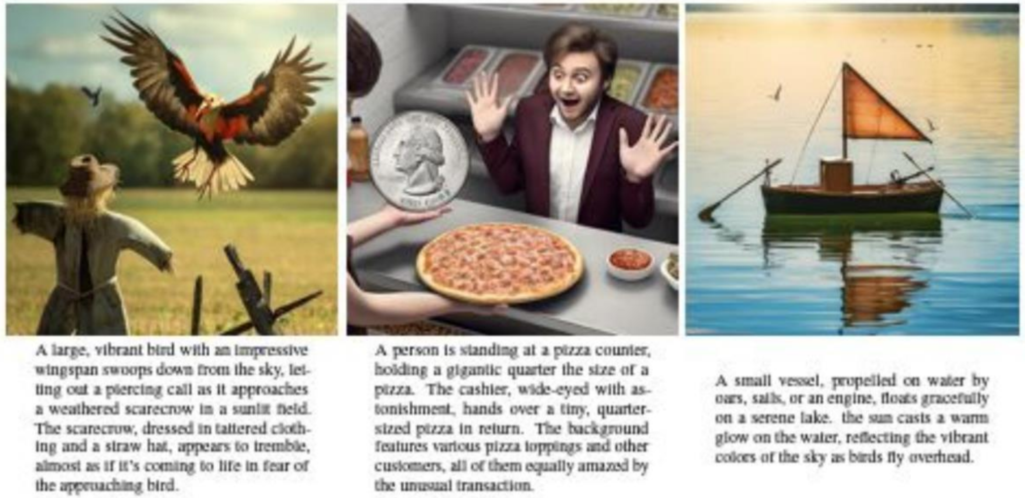
Almost all points are close to decision boundaries (but not classified correctly) due to isoperimetry property in high dim

We provide a theoretical model and a rigorous proof that explains the fact in very similar data setting

The Dimpled Manifold Model of Adversarial Examples in Machine Learning

Final Remarks

- Human is more robust to small perturbations
  - No adv training for human
  - Adv training is slow (can we use std training to get a robust model?)
- DL classifiers only use the class labels as the supervisory information
- More detailed and structured supervisory information for human
  - Patches of images are “Tokenize” to concepts
- More detailed labeling in large scale is possible in the era of MLLM



Thanks

Jian Li 李建
lapordge@gmail.com
Wechat id: lapordge

Relation to properties of KKT points

- Vardi et al. (NeurIPS 2022) and Frei et al. (NeurIPS 2024) show that every KKT point is at most $O(\sqrt{d/k})$-robust for a very similar data distribution (but a $O(\sqrt{d})$-robust solution exists)
- The limiting case (not sure how long one can reach a KKT point). Empirically, some KKT requires very long training time (for certain initializations)
- It is less intuitive what a KKT point look like

Theorem 4.2. *Let $\epsilon, \delta \in (0,1)$. Let $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n \subseteq \mathbb{R}^d \times \{-1,1\}$ be a training set drawn i.i.d. from the distribution $\mathcal{D}_{clusters}$, where $n \geq k \ln^2(d)$. We denote $Q_+ = \{q \in [k] : y^{(q)} = 1\}$ and $Q_- = \{q \in [k] : y^{(q)} = -1\}$, and assume that $\min\left\{\frac{|Q_+|}{k}, \frac{|Q_-|}{k}\right\} \geq c$ for some $c > 0$. Let $\mathcal{N}_{\boldsymbol{\theta}}$ be a depth-2 ReLU network such that $\boxed{\boldsymbol{\theta} = [\mathbf{w}_1, \ldots, \mathbf{w}_m, \mathbf{b}, \mathbf{v}] \text{ is a KKT point of}}$ Problem (2). Provided d is sufficiently large such that $\delta^{-1} \leq \frac{1}{3} d^{\ln(d)-1}$ and $n \leq \min\left\{\sqrt{\frac{\delta}{3}} \cdot e^{d/32}, \frac{\sqrt{\delta}}{3} \cdot d^{\ln(d)/4}, \frac{\epsilon}{4} \cdot d^{\ln(d)/2}\right\}$, with probability at least $1-\delta$ over $\mathcal{S}$, there is a vector $\boxed{\mathbf{z} = \eta \cdot \sum_{j \in [k]} y^{(j)} \boldsymbol{\mu}^{(j)}}$ with $\eta > 0$ and $\|\mathbf{z}\| \leq \mathcal{O}\left(\sqrt{d/c^2 k}\right)$, such that*

a (universal) attack direction

$$\Pr_{(\mathbf{x},y)\sim\mathcal{D}_{clusters}} \underbrace{\left[\operatorname{sign}(\mathcal{N}_{\boldsymbol{\theta}}(\mathbf{x})) \neq \operatorname{sign}(\mathcal{N}_{\boldsymbol{\theta}}(\mathbf{x} - y\mathbf{z}))\right]}_{\text{Attack along direction } y\mathbf{z} \text{ is successful}} \geq 1 - \epsilon\,.$$