清華大學 交叉信息研究院
Institute for Interdisciplinary Information Sciences, Tsinghua University

2024 CSML

Implicit Bias and Adversarial Robustness in 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 Learning Theory

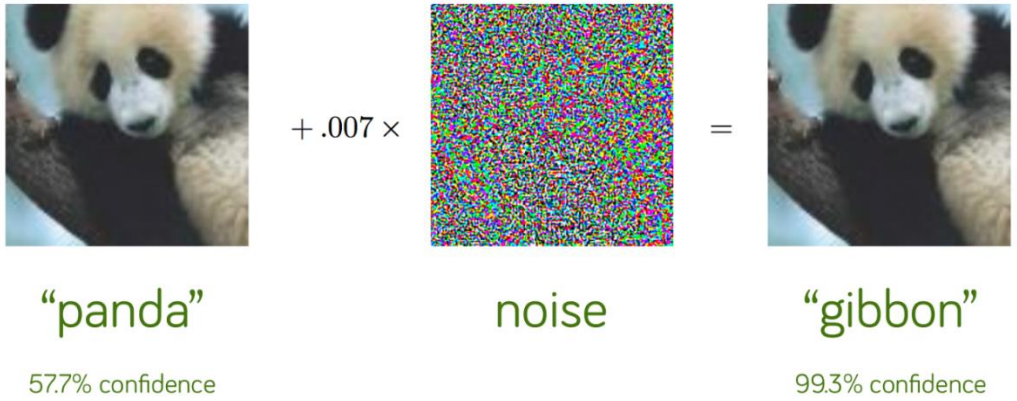
- Tremendous success in practice
- Theory, exciting recent progress (still not so satisfying)

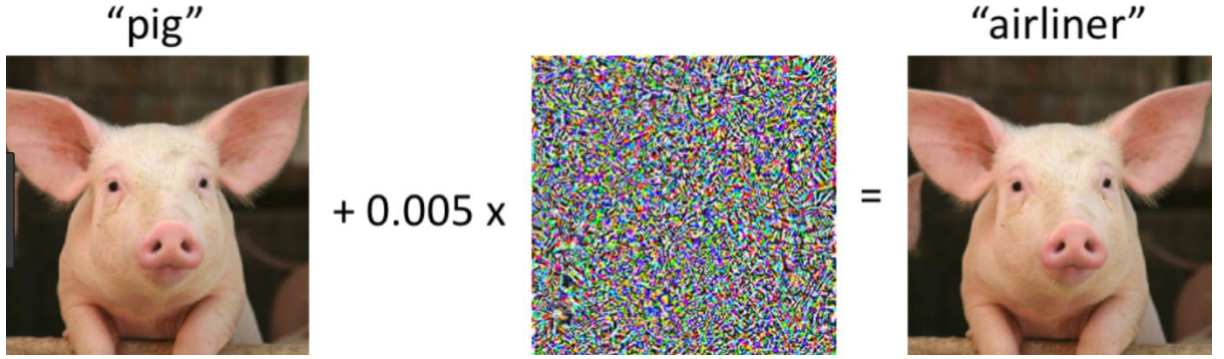
Deep Learning Mysteries

- 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 (not just the network architectures)
  - Optimization
  - Algorithm-dependent generalization
  - **Implicit bias** (towards local/global min with interesting properties)
- Inductive bias
  - Why CNN works so well for image data?
- Many useful tricks
  - Dropout, batchnorm, layernorm, initialization
- **Existence of Adversarial Examples**

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
- Geometrically, every training sample (as well as testing sample) is very close to the decision boundary.
- Very intriguing phenomena (beyond safety issue)
- Robust decision boundary exists (Human is a robust classifier)

Outline

- Implicit Bias
- Margin Maximization for DNN
- Margin and Robustness
- Feature Averaging
- Feature Averaging leads to Nonrobust Solutions
- Relation to Existing Models
  - Dimpled Manifold, Nonrobust Features, etc.

Implicit Bias

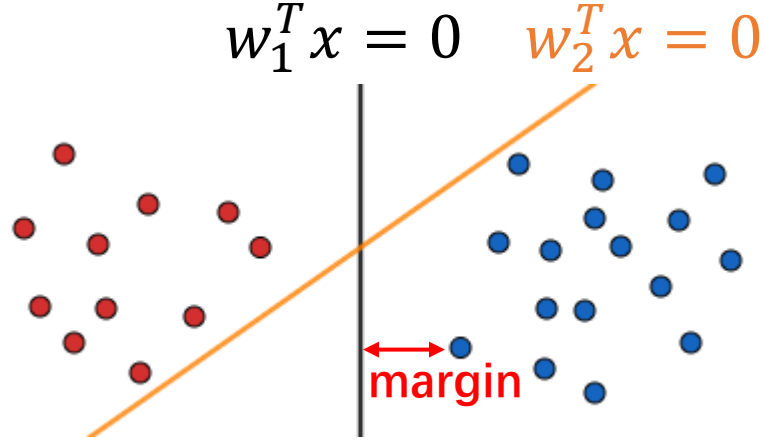
- The optimization algorithm may **implicitly bias** the solutions to global/local minima with special properties.
  - Implicit bias is particularly important in learning deep neural networks as "it introduces **effective capacity control** not directly specified in the objective" [Gunasekar et al. 18]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

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.

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

"find the solutions with smaller norm"

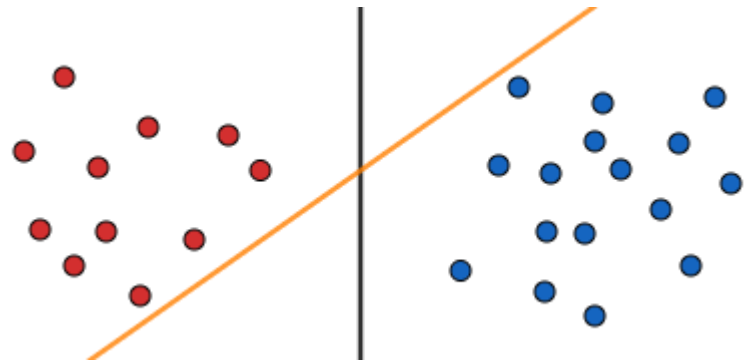
Theorem (Rosset et al., 2004, informal).
When λ is small, the global minimizer of $\mathcal{L}_\lambda(w)$ is close to the SVM solution.

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

max-margin linear classifier
(solving the unconstrained optimization = the constrained program)

adapted 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.

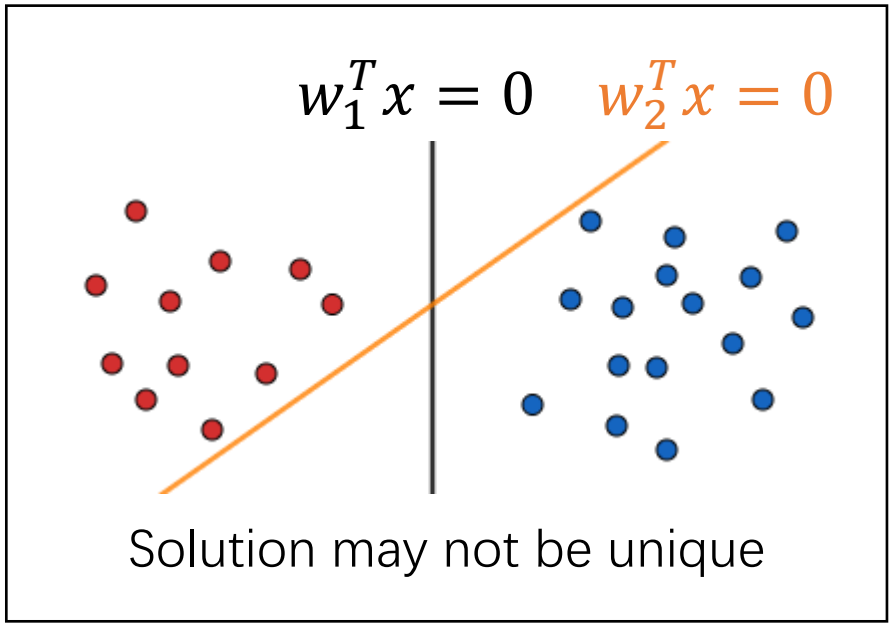
Does GD have a similar "implicit bias" on deep neural nets?

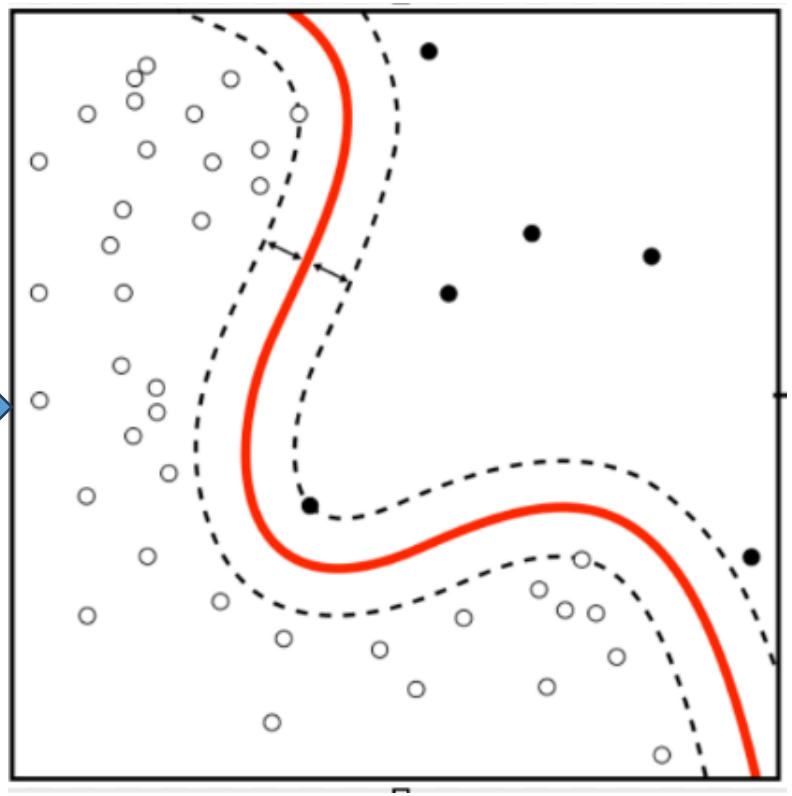
Outline

- Implicit Bias
- Margin Maximization for DNN
- Margin and Robustness
- Feature Averaging
- Feature Averaging leads to Nonrobust Solutions
- Relation to Existing Models
  - Dimpled Manifold, Nonrobust Features, etc.

Margin Maximization for DNN?

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



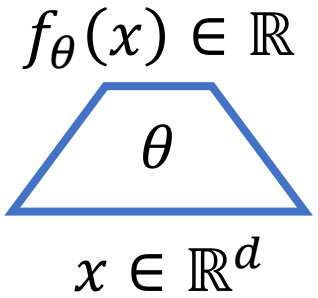


How to define margin for DNN:

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

Margin for Homogeneous Neural Nets

"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)



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

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].

- Lyu, **L**. 2020. ICLR 2020 oral.

Implicit Bias: Margin Maximization

- 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

Smoothed normalized margin (change min to softmin)

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

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

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* $t > t_0$,

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

Extension to certain non-homogeneuous NN by Chatterji, Long, Bartlett.

Implicit bias: Margin Maximization

Max-Margin Problem: (P)

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

Classical SVM

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

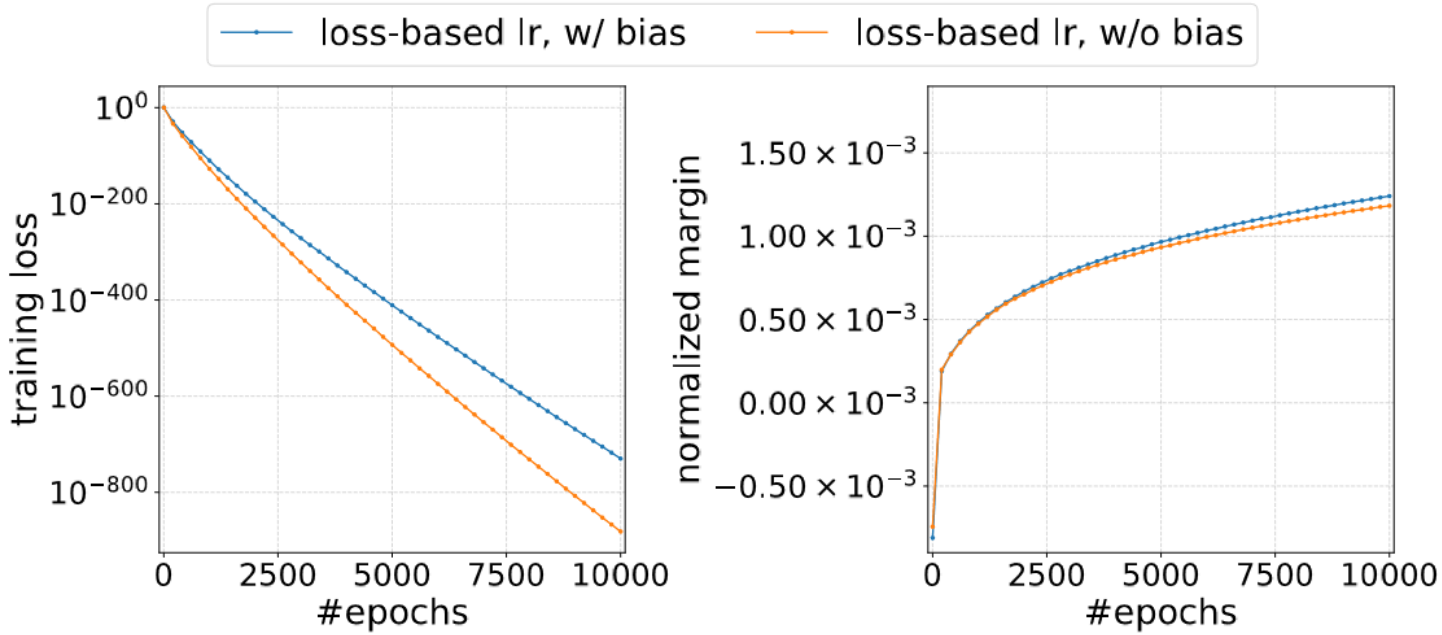
Theorem (Lyu, L. 2020., Ji, Telgarsky 2020.) **The direction $\widehat{\boldsymbol{\theta}}$ converges and for the limit direction of $\widehat{\boldsymbol{\theta}}$, $\widehat{\boldsymbol{\theta}}/q_{\min}(\widehat{\boldsymbol{\theta}})^{1/L}$ is a KKT point of (P).**

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

1. $\boldsymbol{\theta} - \sum_{n=1}^{N} \lambda_n \boldsymbol{h}_n = \mathbf{0}$ for some $\boldsymbol{h}_1, \ldots, \boldsymbol{h}_N$ 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



(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

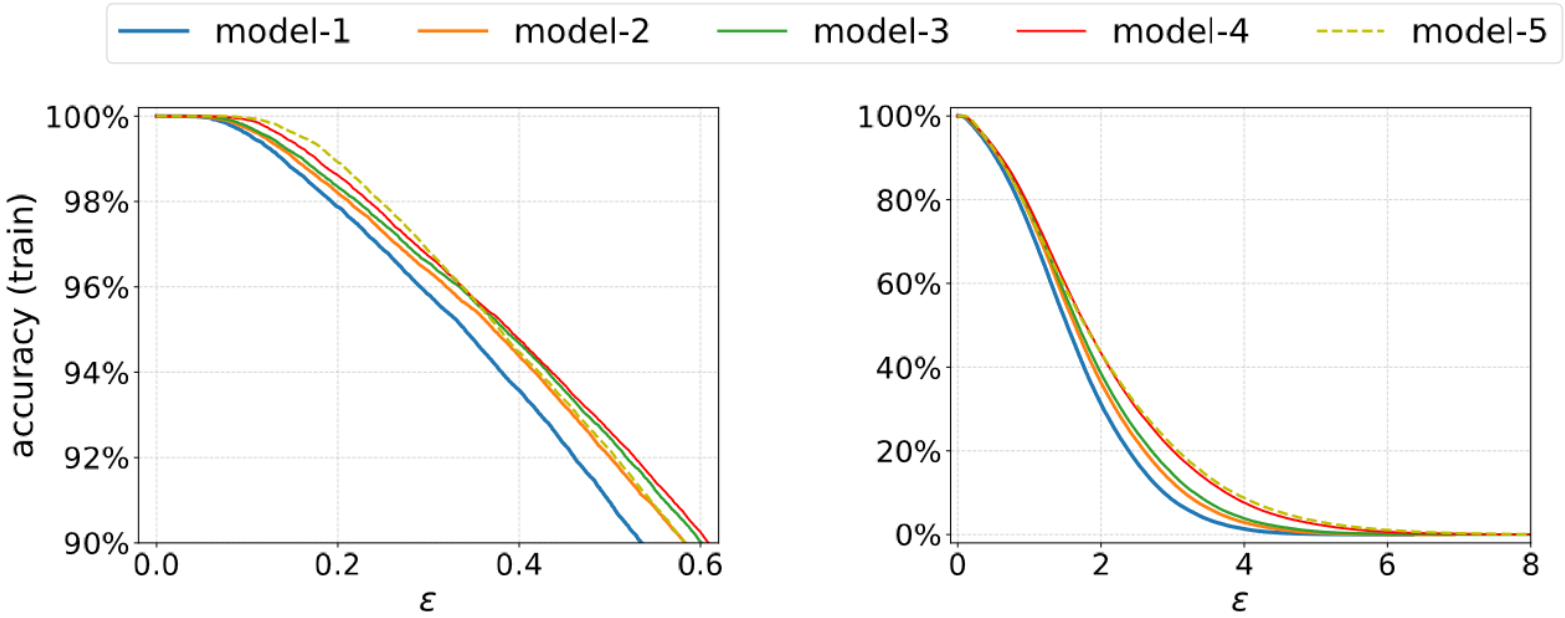
Outline

- Implicit Bias
- Margin Maximization for DNN
- Margin and Robustness
- Feature Averaging
- Feature Averaging leads to Nonrobust Solutions
- Relation to Existing Models
  - Dimpled Manifold, Nonrobust Features, etc.

Robustness

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

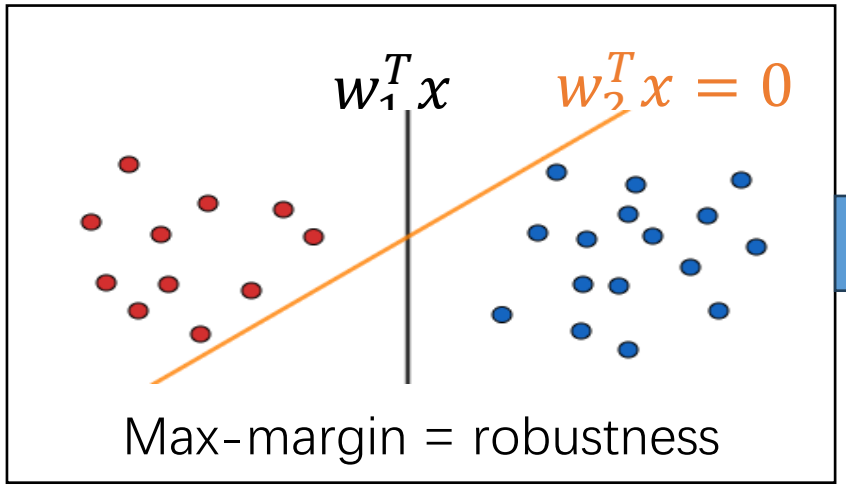
model name	number of epochs	train loss	normalized margin
model-1	38	$10^{-10.04}$	5.65×10^{-5}
model-2	75	$10^{-15.12}$	9.50×10^{-5}
model-3	107	$10^{-20.07}$	1.30×10^{-4}
model-4	935	$10^{-120.01}$	4.61×10^{-4}
model-5	10000	$10^{-881.51}$	1.18×10^{-3}

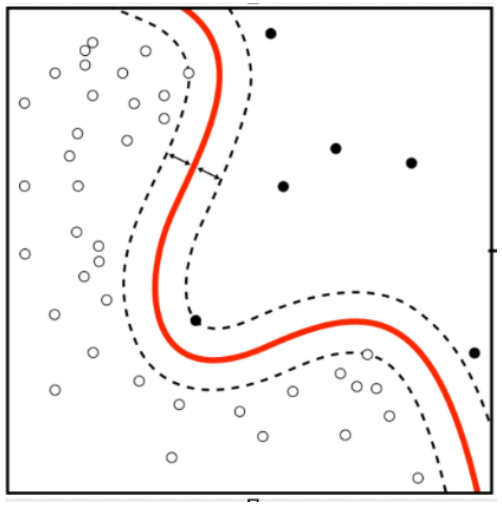


Hence, training longer is slightly useful in improving the robustness.

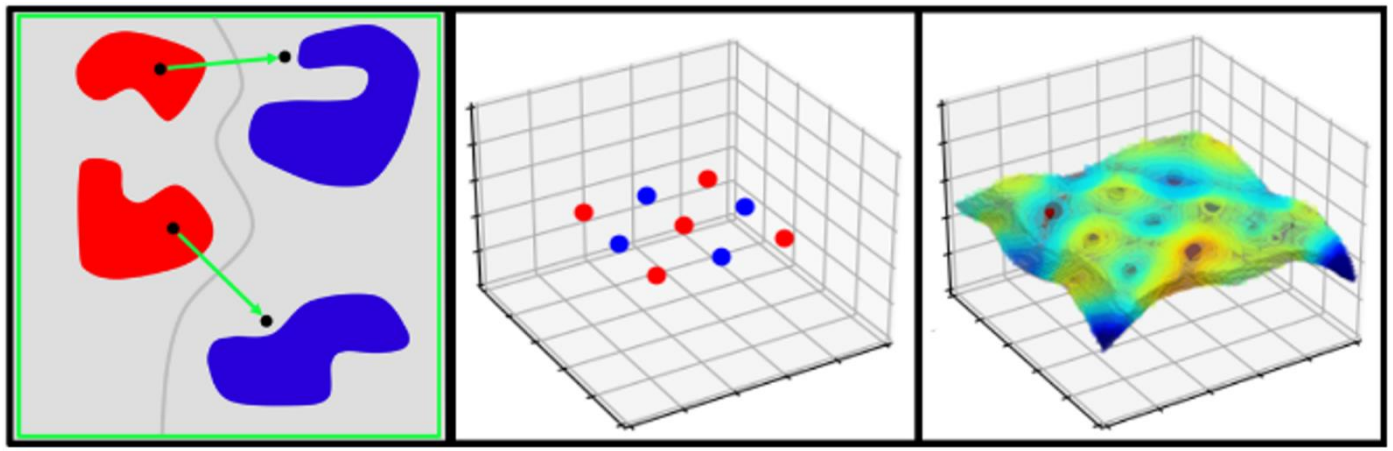
Margin and Robustness

It seems that we have solved the robustness problem (via margin maximization)..But of course we haven't!





This picture may be misleading (especially in high dim)



The Dimpled Manifold Model of Adversarial Examples in Machine Learning

Outline

- Implicit Bias
- Margin Maximization for DNN
- Margin and Robustness
- Feature Averaging
- Feature Averaging leads to Nonrobust Solutions
- Relation to Existing Models
  - Dimpled Manifold, Nonrobust Features, etc.

Implicit Bias of GD

Double-edged sword of GD (Frei, Vardi, Bartlett, Srebro 24)

- On one hand, GD leads to good standard accuracy
- On the other hand, the KKT properties (the implicit bias of GD) force the network to find non-robust solution
  - But KKT properties are abstract and hold only for limiting case

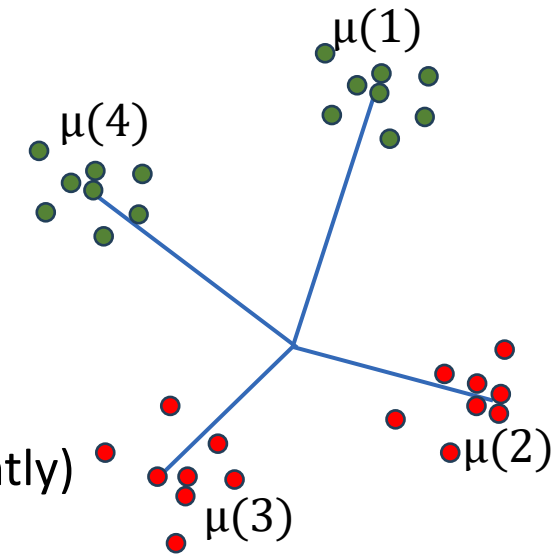
- We perform a more intuitive, fine-grained, and finite time analysis of GD process:
- A new form of implicit bias: **Feature Averaging**
  - While there exist many discriminative features capable of classifying data, GD tends to learn the average/combination of these features
  - One of major causes of nonrobustness

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

Theoretical Setup

Data distribution:

- D_{binary} on $R^d \times \{-1, 1\}$ that consists of k clusters
  - 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:

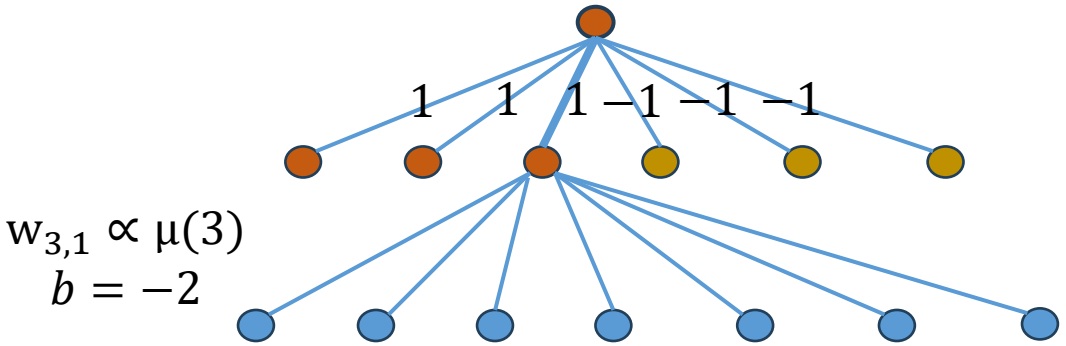
- For simplicity, fix the second layer

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

- Loss function (logistic loss): $\mathcal{L}(\boldsymbol{\theta}) := \frac{1}{n} \sum_{i=1}^{n} \ell\left(y_i f_{\boldsymbol{\theta}}\left(\mathbf{x}_i\right)\right) \qquad \ell(q) = \log\left(1 + e^{-q}\right)$
- 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): $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla \mathcal{L}(\boldsymbol{\theta}_t)$

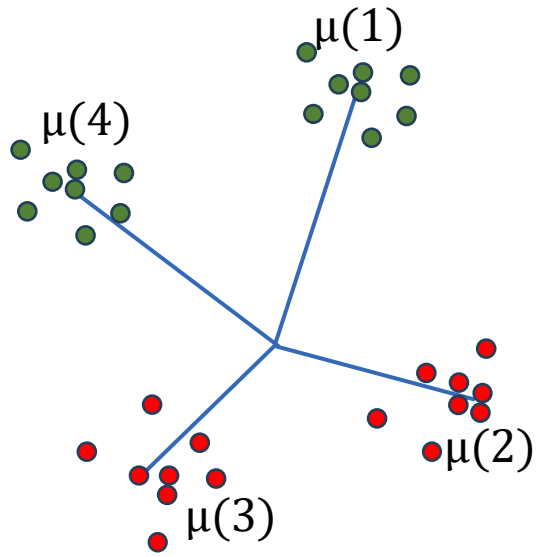
Robust solution exists

- It is easy to show a **robust solution exists** with robust radius $O(\sqrt{d})$
  - Let each neuron deal with one cluster
  - 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



GD learns Average Features

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

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

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**:

$$\lambda_{s,r,j}^{(T)} = \tilde{\Omega}(1), \frac{\lambda_{s,r,j_1}^{(T)}}{\lambda_{s,r,j_2}^{(T)}} = 1 \pm o(1), \forall s \in \{-1,+1\}, r \in [m], j, j_1, j_2 \in \mathcal{J}_s$$

Large coeffs for the same class

Large coeffs are almost the same

$$\lambda_{s,r,j}^{(T)} = o(1), \forall s \in \{-1,+1\}, r \in [m], j \in \mathcal{J}_{-s}$$

Other coeffs are negligible

GD learns Average Features

The theorem resolves the conjecture proposed by Min and Vida, ICML 2024 (under slightly different setting)

$$F(\boldsymbol{x}) = \sqrt{K_1}\sigma(\langle \bar{\boldsymbol{\mu}}_+, \boldsymbol{x}\rangle) - \sqrt{K_2}\sigma(\langle \bar{\boldsymbol{\mu}}_-, \boldsymbol{x}\rangle), \quad (3)$$

where $\bar{\boldsymbol{\mu}}_+ = \frac{1}{\sqrt{K_1}}\sum_{k=1}^{K_1}\boldsymbol{\mu}_k$ and $\bar{\boldsymbol{\mu}}_- = \frac{1}{\sqrt{K_2}}\sum_{k=K_1+1}^{K}\boldsymbol{\mu}_k$

Average features

Conjecture 1. *Suppose that the intra-subclass variance $\alpha > 0$ is sufficiently small, that one has a training dataset of sufficiently large size, and that we run gradient flow training on $f_p(\boldsymbol{x};\boldsymbol{\theta}), \boldsymbol{\theta} = \{\boldsymbol{w}_j, v_j\}_{j=1}^h$ of sufficiently large width h for sufficiently long time T, starting from random initialization of the weights with a sufficiently small initialization scale. If $p = 1$, then we have $\inf_{c>0}\sup_{\boldsymbol{x}\in\mathbb{S}^{D-1}}|cf_p(\boldsymbol{x};\boldsymbol{\theta}(T)) - F(\boldsymbol{x})| \ll 1$; If $p \in [3,\bar{p})$ for some $\bar{p} > 3$, then we we have $\inf_{c>0}\sup_{\boldsymbol{x}\in\mathbb{S}^{D-1}}|cf_p(\boldsymbol{x};\boldsymbol{\theta}(T)) - F^{(p)}(\boldsymbol{x})| \ll 1$.*

Outline

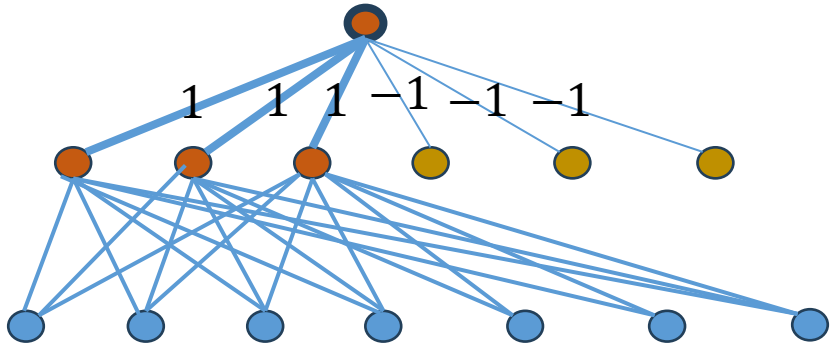
- Implicit Bias
- Margin Maximization for DNN
- Margin and Robustness
- Feature Averaging
- Feature Averaging leads to Nonrobust Solutions
- Relation to Existing Models
  - Dimpled Manifold, Nonrobust Features, etc.

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 $\delta-$robust accuracy for any robust radius $\delta = \omega(\sqrt{d/k})$

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

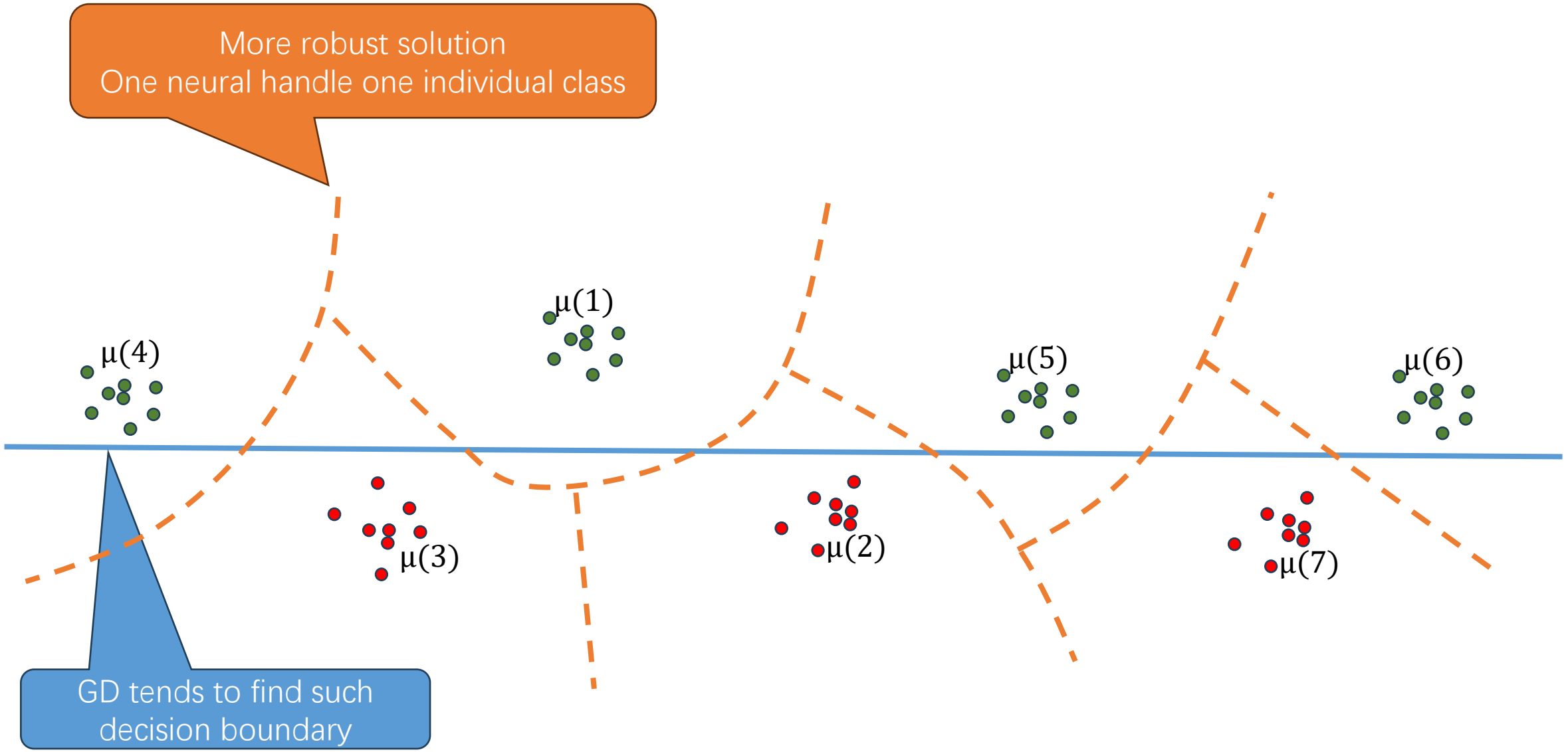
Intuition: for average features, most same-class neurons will be activated, resulting a much larger gradient norm (even though the margin $y_i f(x_i)$ is similar to that in a robust solution)



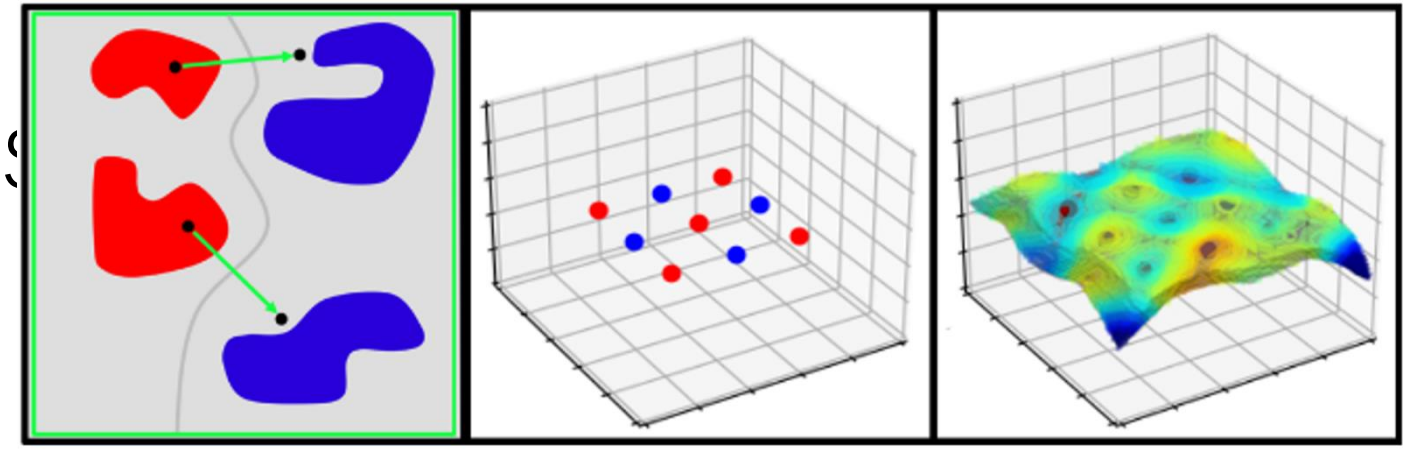
$$\boldsymbol{w}_{s,r}^{(t)} = \boldsymbol{w}_{s,r}^{(0)} + \sum_{j\in\mathcal{J}_+} \lambda_{s,r,j}^{(t)} \boldsymbol{\mu}_j + \sum_{j\in\mathcal{J}_-} \lambda_{s,r,j}^{(t)} \boldsymbol{\mu}_j + \sum_{i\in[N]} \sigma_{s,r,i}^{(t)} \boldsymbol{\xi}_i$$

large small

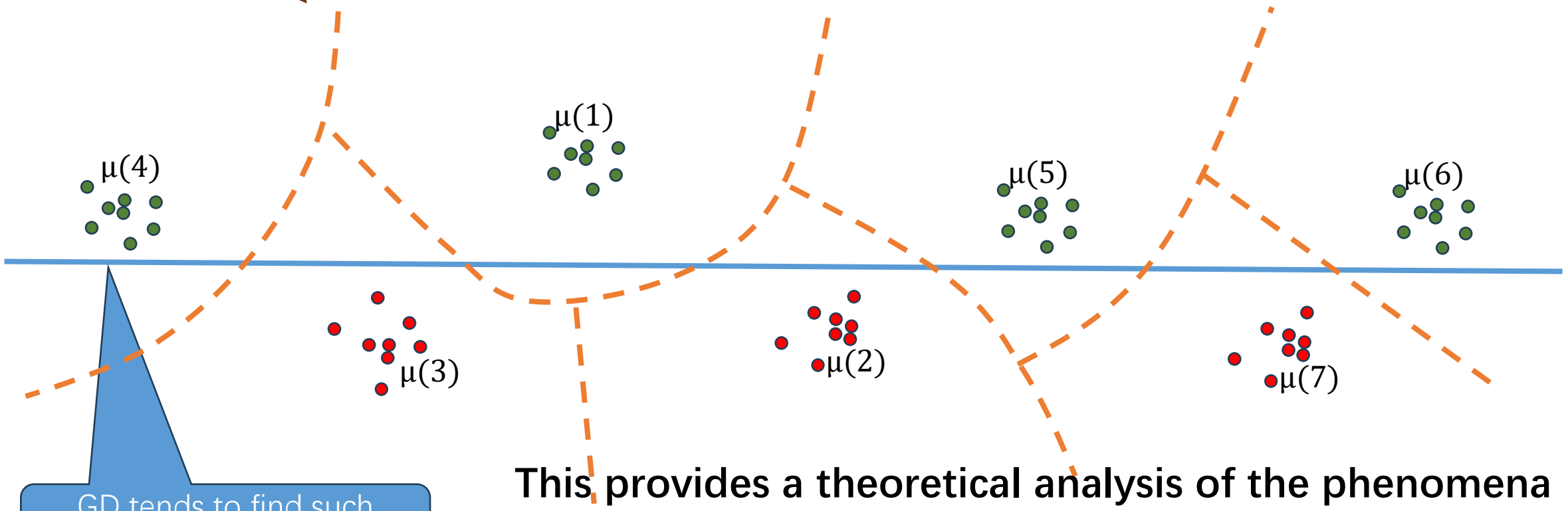
Robust and Nonrobust Solutions



Robust and Nonrobus



More robust solution
One neural handle one individual class



GD tends to find such decision boundary

This provides a theoretical analysis of the phenomena described in dimpled manifold hypothesis in our setting

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.

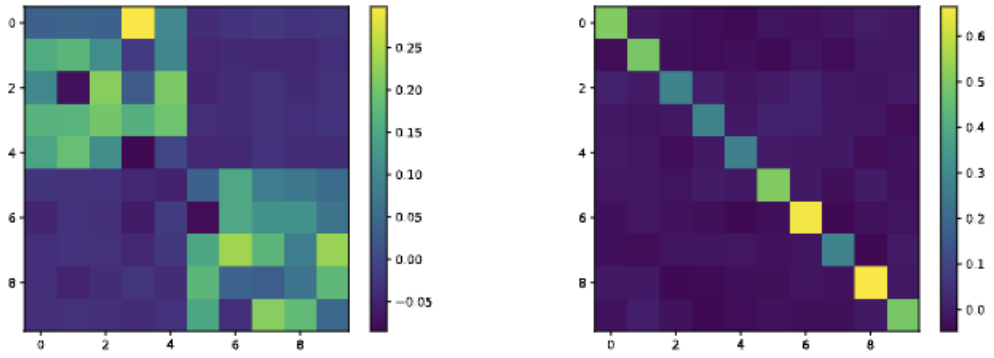


Figure 1: **Illustration of Feature Averaging and Feature Decoupling** .

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

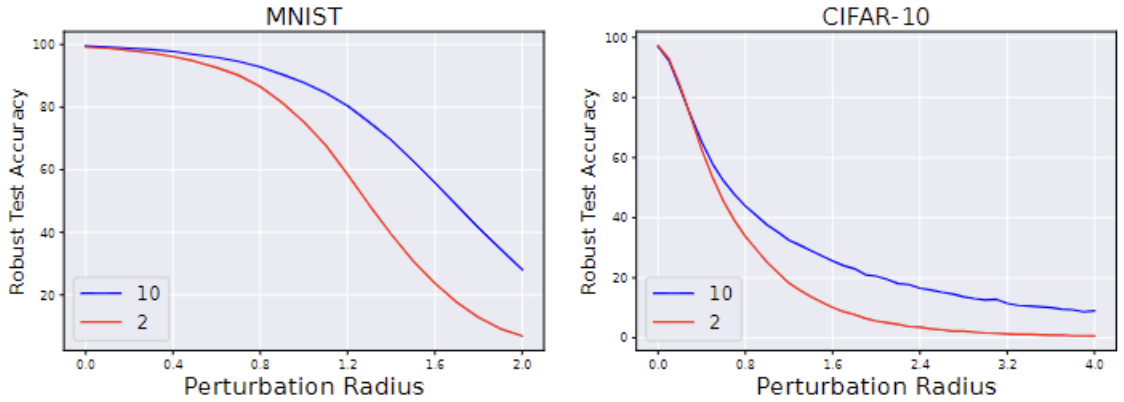


Figure 2: **Robustness Improvement on MNIST and CIFAR10** .

Outline

- Implicit Bias
- Margin Maximization for DNN
- Margin and Robustness
- Feature Averaging
- Feature Averaging leads to Nonrobust Solutions
- Relation to Existing Models
  - Dimpled Manifold, Nonrobust Features, etc.

Connection to Simplicity Bias

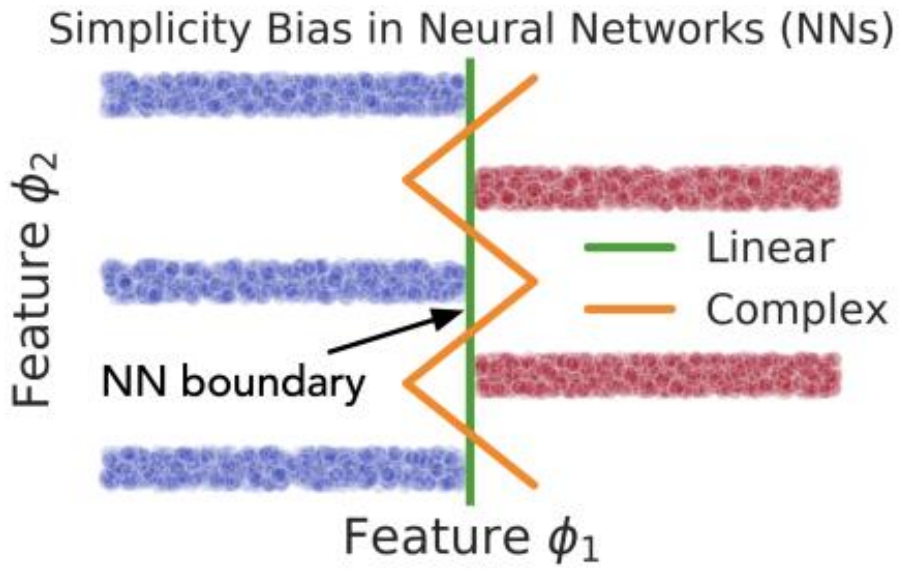
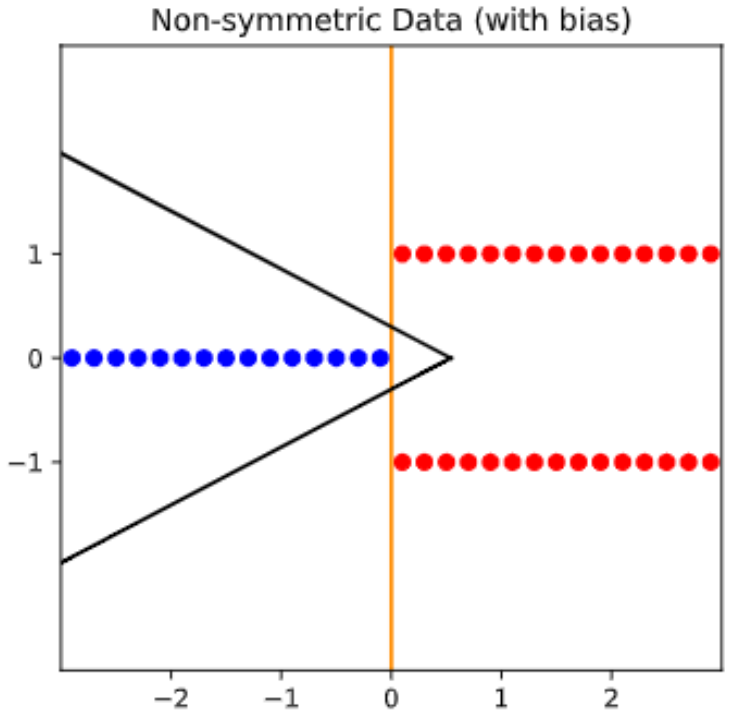


Figure 1: Simple vs. complex features

Shah et al. [2020]

The Pitfalls of Simplicity Bias in Neural Networks

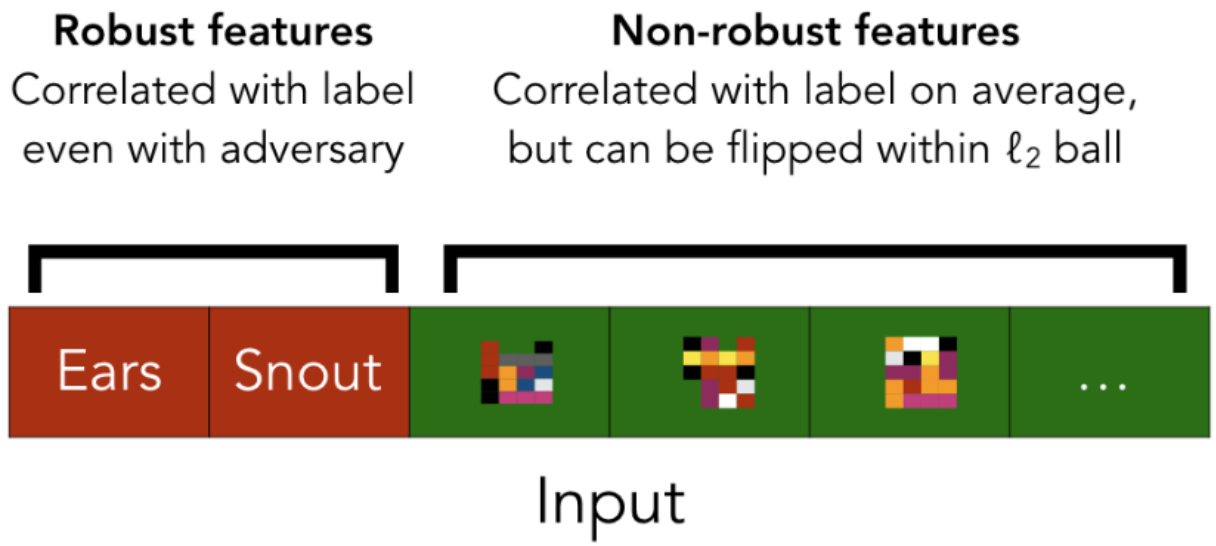


black: max-margin
orange: linear

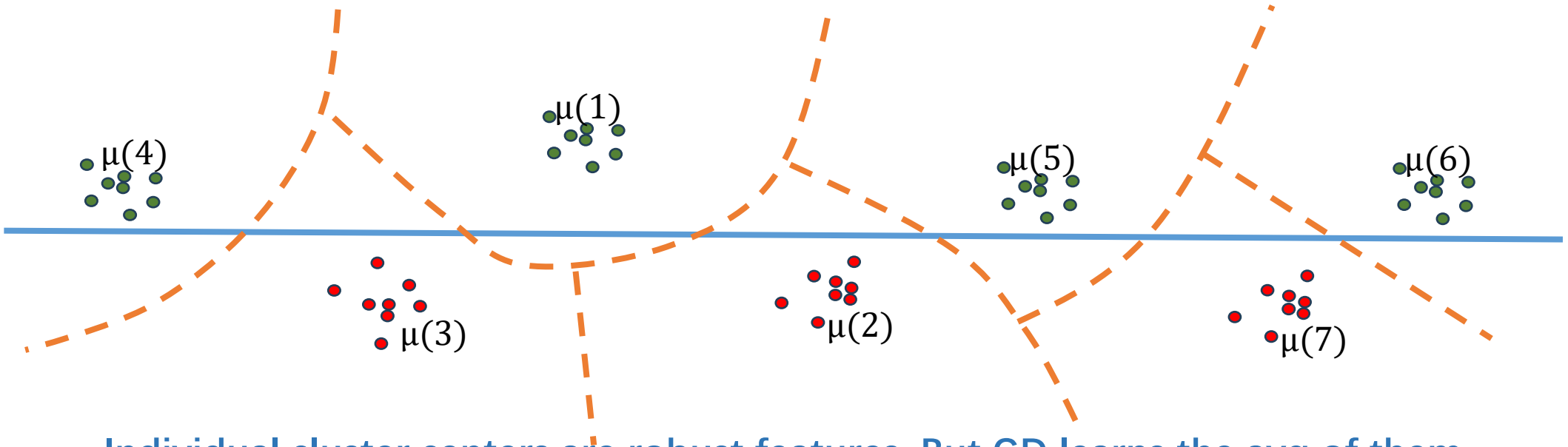
One can show GD on a 2-layer NN (with small init) finds a linear classifier theoretically.
(A linear classifier only maximize the margin locally. Clearly it is not a global margin maximizer)

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

Connection to Nonrobust Features



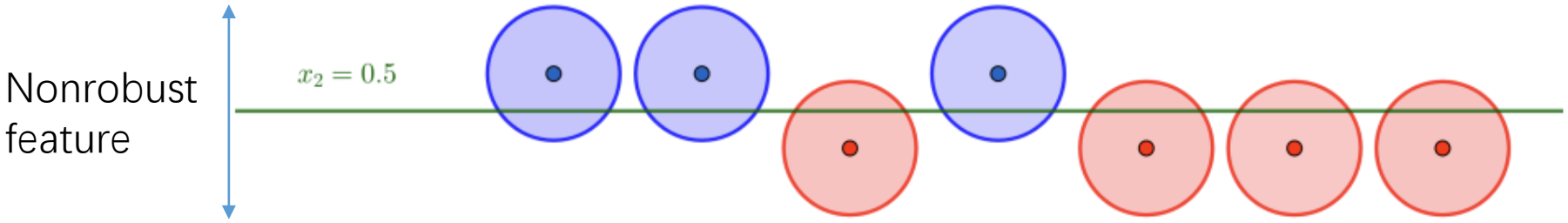
Ilyas et al. Adversarial Examples Are Not Bugs, They Are Features



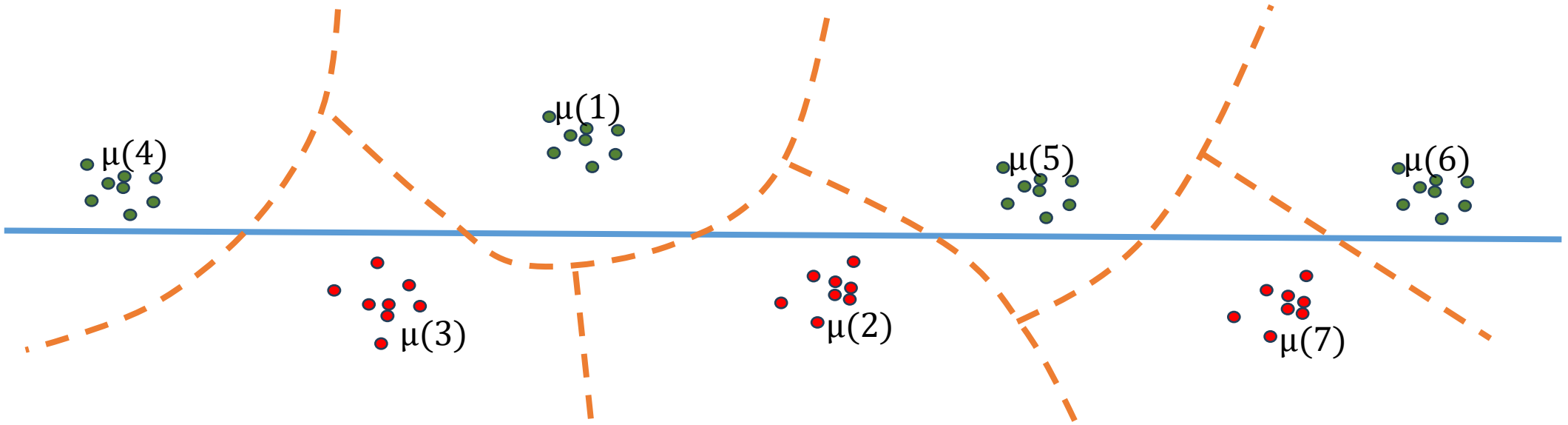
Individual cluster centers are robust features. But GD learns the avg of them, which is a nonrobust feature

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)



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



Final Remarks

Detailed feature-level supervisory label is useful

- We also show if one is provided detailed feature level label, a similar 2-layer NN can learn feature decoupled solutions (which is more robust)

- 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?)
  - More detailed and structured supervisory information for human
  - Such labeling in large scale is possible in the era of multi-model LLM

Thanks

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

Robustness

- Robustness

$$R_{\boldsymbol{\theta}}(\boldsymbol{z}) := \inf_{\boldsymbol{x}' \in X} \{\|\boldsymbol{x} - \boldsymbol{x}'\| : (\boldsymbol{x}', y) \text{ is misclassified}\}$$

- 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