Feature Averaging: An Implicit Bias of Gradient Descent Leading to Non-Robustness in Neural Networks[1,2]

Binghui Li

Zhixuan Pan

Kaifeng Lyu

Jian Li

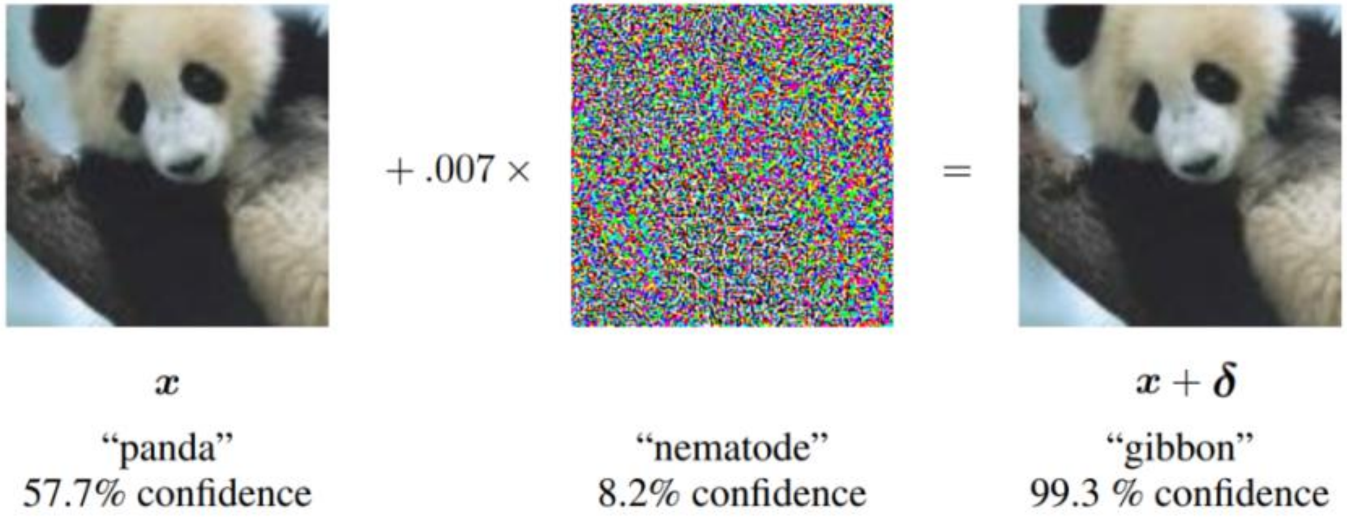
---

[1]This work has been accepted by **ICLR 2025**, where the first two authors have equal contributions and the last author is the corresponding author.

[2]Our full paper can be found at https://arxiv.org/abs/2410.10322 .

Adversarial Examples

- Although deep neural networks have achieved remarkable success in practice, it is well-known that modern neural networks are vulnerable to adversarial examples.
- Specifically, for a given image x, an indistinguishable small but adversarial perturbation δ is chosen to fool the classifier f to produce a wrong class using $f(x+\delta)$ [Szegedy et al, 2013].

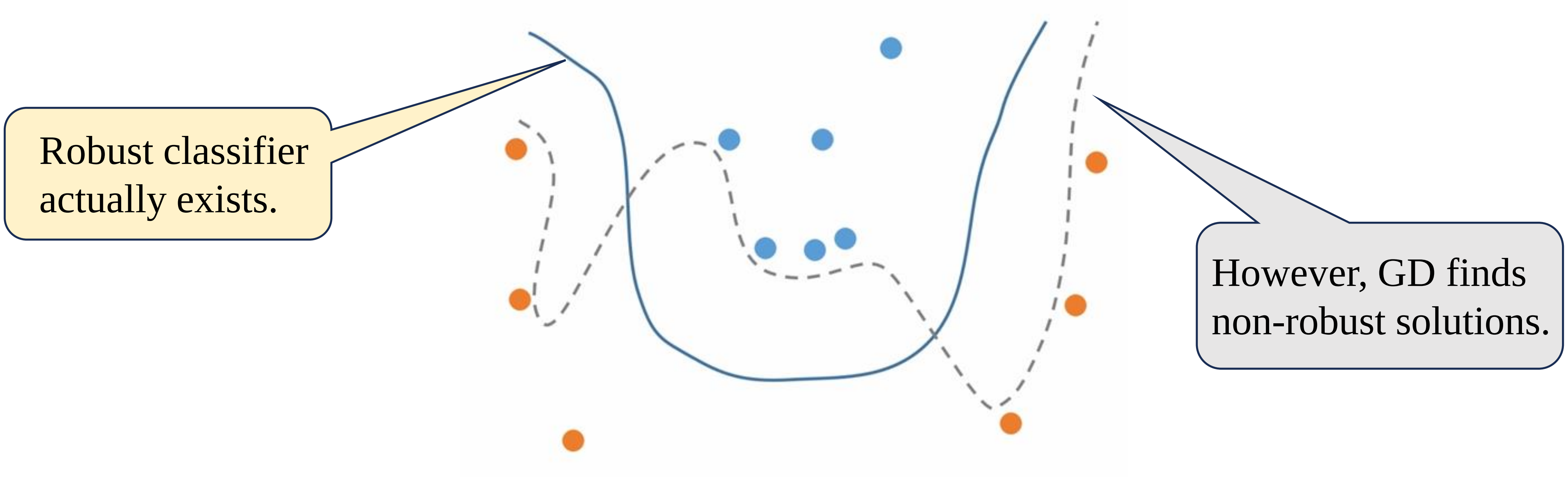


An Instance for Adversarial Example

Question

Our Fundamental Theoretical Questions :

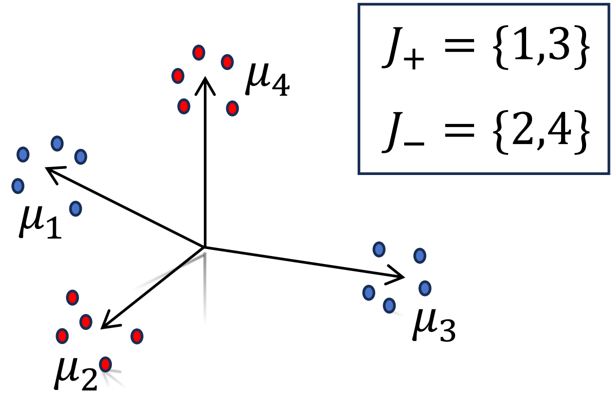
*Why do neural networks trained by **gradient descent algorithm** converge to the **non-robust solutions** that fail to classify **adversarial examples**?*



Data Distribution

- Data distribution D_{binary} on $\mathbb{R}^d \times \{-1,1\}$ that consists of ***k* clusters**:
  - for each cluster, it corresponds to a cluster feature vector μ_i ($i \in [k]$) ;
  - μ_i for all $i \in [k]$ are orthogonal and $\|\mu_i\|_2 = \Theta(\sqrt{d})$;
  - Suppose that total ***k* clusters** can be divide into two disjoint classes with index sets J_+ and J_- that correspond to positive class and negative class, respectively;
  - positive and negative clusters are balanced: $\exists c \geq 1, c^{-1} \leq \frac{|J_+|}{|J_-|} \leq c$.
- An instance (x, y) sampled from cluster i:
  - label $y = 1$ if $i \in J_+$ and $y = -1$ if $i \in J_-$;
  - data input $x = \mu_i + \xi$,

  where random noise $\xi \sim N(0, \sigma^2 I_d)$ and $\sigma = \Theta(1)$.



An example for $k = 4, c = 1$

The similar data distribution is analyzed in [Frei et al, 2024].

Learner Model: Two-Layer ReLU Network

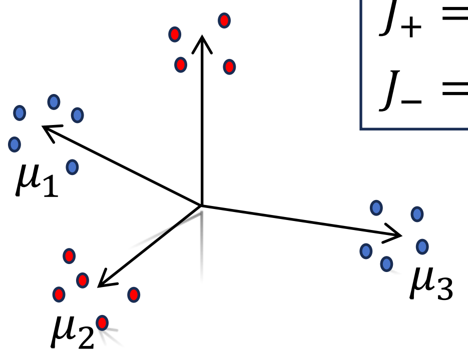
- **Two-layer ReLU network:** for simplicity, we fix the second layer.

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

  where $\theta = \{w_{s,r}, b_{s,r}\}_{(s,r)\in\{1,-1\}\times[m]}$ are trainable parameters.
- **Loss function:** we apply logistic loss as $L(\theta) := \frac{1}{n}\sum_{i=1}^{n} l(y_i f_\theta(x_i))$, where $l(z) := \log(1 + e^{-z})$.
- **Initialization:** $w_{s,r}^{(0)} \sim N(0, \sigma_w^2 I_d)$, $\sigma_w^2 = \frac{1}{d}$ and $b_{s,r}^{(0)} \sim N(0, \sigma_b^2)$, $\sigma_b^2 = \frac{1}{d^2}$.
- **Gradient descent algorithm:** $\theta_{t+1} = \theta_t - \eta\nabla_\theta L(\theta_t)$ with small learning rate $\eta = \Theta(\frac{1}{\sqrt{d}})$.

There Exists the Robust Solution!

- Indeed, it is easy to show a robust solution exists with **robust radius $\boldsymbol{O(\sqrt{d})}$:**
  - Let each neuron deal with one cluster;
  - Use the bias term to filter out intra/inter cluster noise.



$J_+ = \{1,3\}$

$J_- = \{2,4\}$

$$f_{robust}(x) = \sum_{j\in J_+} \underbrace{ReLU(\langle \mu_j, x\rangle + b_j^+)}_{\text{deal with positive cluster } j} - \sum_{l\in J_-} \underbrace{ReLU(\langle \mu_l, x\rangle + b_l^+)}_{\text{deal with negative cluster } l}$$

An example for $k = 4, c = 1$

$\forall i \neq j, \|\mu_i - \mu_j\|_2 = \Theta(\sqrt{d})$

f_{robust} achieves optimal robustness.

GD Provably Learns Averaged Features

- **Lemma** (Weight Decomposition). During training, we can decompose the weight $w_{s,r}^{(t)}$ as linear combination of the features (and some noise):

$$\boxed{w_{s,r}^{(t)} = w_{s,r}^{(0)} + \sum_{j \in J_+} \lambda_{s,r,j}^{(t)} \mu_j + \sum_{j \in J_-} \lambda_{s,r,j}^{(t)} \mu_j + \sum_{i \in [n]} \sigma_{s,r,i}^{(t)} \xi_i.}$$

- **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:
  - The model achieves perfect standard accuracy: $\mathbb{P}_{(x,y)\sim D_{binary}}\left[\mathrm{sgn}\left(f_{\theta^{(T)}}(x)\right) = y\right] = 1 - o(1)$.
  - GD learns averaged features:

$$\underbrace{\lambda_{s,r,j}^{(T)} \geq \Omega(1),}_{\text{Large coeffs for the same class}} \quad \underbrace{\lambda_{-s,r,j}^{(T)} \leq o(1),}_{\text{Small coeffs for the other class}} \quad \underbrace{\frac{\lambda_{s,r,j}^{(T)}}{\lambda_{s,r,k}^{(T)}}}_{\text{No large coeff is much than others}} \leq O(1), \quad \forall s \in \{-1,1\}, r \in [m], j \neq k \in J_s.$$

Intuitively, it approximately satisfies:

$$w_{s,r} \propto \sum_{j \in J_s} \mu_j \,, \forall (s,r) \in \{-1,1\} \times [m]$$

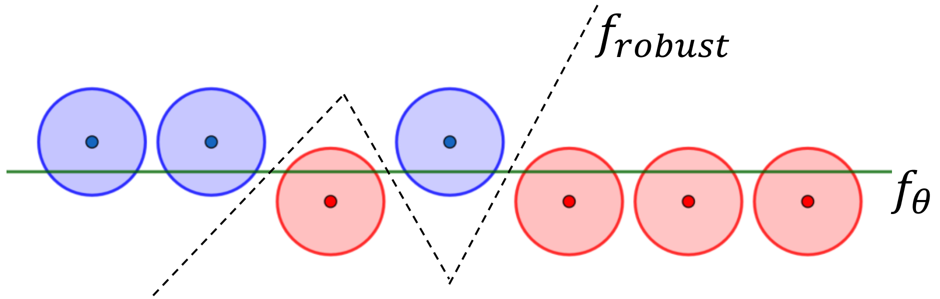
Averaged Features are Non-robust Features

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

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

Intuition: for averaged features, the model approximately degenerates into a two-neuron network as follows,

$$f_\theta(x) \approx C(ReLU(\underbrace{\langle \textstyle\sum_{j\in J_+} \mu_j, x\rangle + b_+}_{\text{deal with all positive clusters}}) - ReLU(\underbrace{\langle \textstyle\sum_{j\in J_-} \mu_j, x\rangle + b_-}_{\text{deal with all negative clusters}}))$$



In fact, the attack can be chosen as $\varepsilon \propto -\sum_{j\in J_+} \mu_j + \sum_{j\in J_-} \mu_j$

Detailed Feature-Level Supervisory Label

- One can show if one is provided detailed feature level label, some two-layer ReLU network can learn feature-decoupled solutions, which is provably more robust.

Theorem (Multiple-Info Helps Learning Feature-Decoupled Solutions). By given all cluster information for each data point, we can apply the standard gradient descent algorithm to solve the corresponding k-classification task, and we will derive the following multiple classifier $F(x) = (f_1,\ldots,f_k): \mathbb{R}^d \to \mathbb{R}^k$, where $f_i(x) \coloneqq ReLU(\langle w_i, x\rangle)$, which satisfies

- $w_i^{(t)} = w_i^{(0)} + \sum_{j\in[k]} \lambda_{i,j}^{(t)} \mu_j + \sum_{l\in[n]} \sigma_{i,l}^{(t)} \xi_l$
- After $T = \Theta(poly(d))$, it holds that: $\lambda_{i,i}^{(T)} = \Omega(1), \lambda_{i,j}^{(T)} = o(1), \forall i \in [k], j \in [k]\backslash\{i\}$.

- Comments: Human is more robust to small perturbations.
  - *No adv training for human.*
  - *Adv training is slow (can we used 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 LLMs.*

Real-World Experiments

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

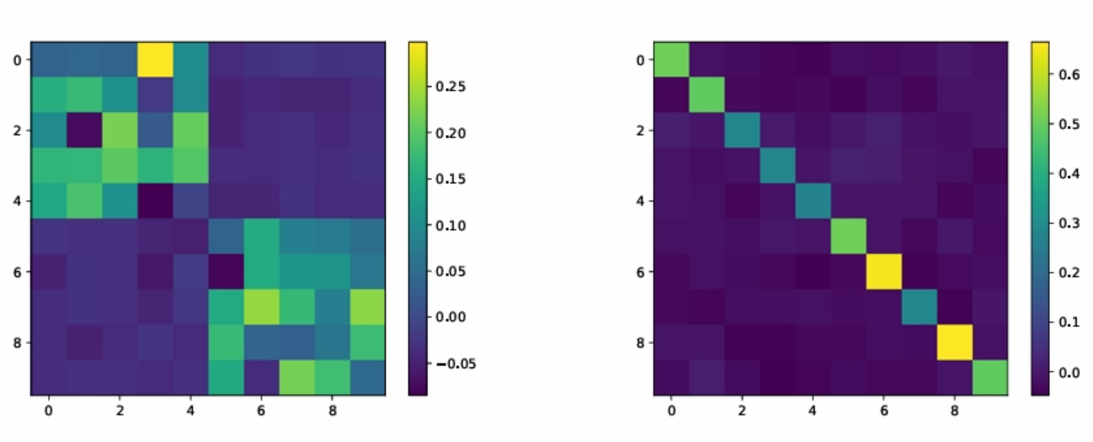


Figure 1: **Illustration of Feature Averaging and Feature Decoupling** .

We create binary classification tasks from the MNIST and CIFAR10 datasets:

- Red: binary classifier trained by 2-classification task.
- Blue: binary classifier trained by 10-classification task.

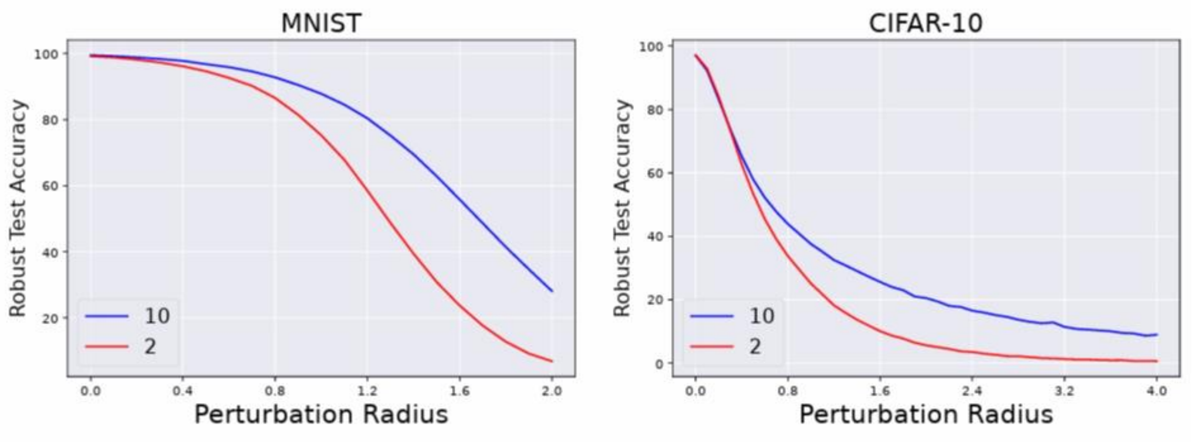
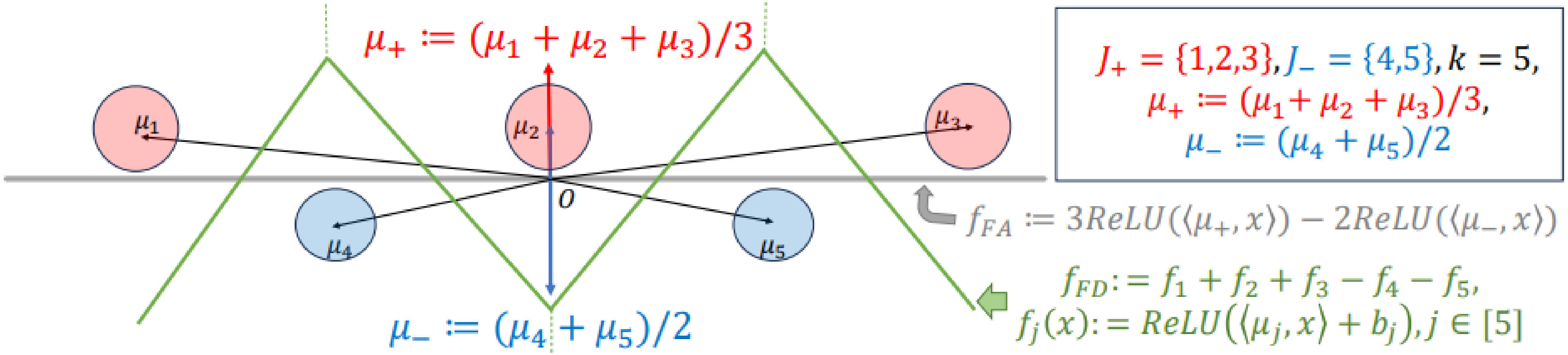


Figure 2: **Robustness Improvement on MNIST and CIFAR10** .

Take-Home Messages

- **Message I:** Adversarial examples may stem from averaged features learned by GD.
- **Message II:** More detailed/ structured supervisory information helps achieving models with better robustness.



Thanks for listening!

My Homepage

Our Full Paper