FEATURE AVERAGING: AN IMPLICIT BIAS OF GRADIENT DESCENT LEADING TO NON-ROBUSTNESS IN NEURAL NETWORKS

DL THEORY PRESENTATION

유병호, 하상범

POSTECH

December 10, 2025

Part I

INTRODUCTION & PROBLEM SETUP

INTRODUCTION

- In class, we learned that a two-layer ReLU network trained by gradient descen tends to learn the *feature directions* of the data.
- This paper goes one step further: it shows that gradient descent does not simply learn individual features, but actually performs *feature averaging* across all clusters within the same label.
- In this talk, we will examine how this averaging phenomenon emerges naturally from the training dynamics.

PROBLEM SETUP

We need two components to fully specify the learning problem:

- **Input & Output Data Distribution:** Defines how labeled samples (x, y) are generated from an underlying cluster structure.
- **Neural Network Learner:** Specifies the architecture and training dynamics of the model used to learn from the data.

Problem Setup

Input & Output Data Dsitrubtion

Definition 2.1 (Multi-Cluster Data Distribution)

Given k vectors $\mu_1, \ldots, \mu_k \in \mathbb{R}^d$, called the cluster features, and a partition of $[k]$ into two disjoint sets $J_\pm = (J_+, J_-)$, we define $D(\{\mu_j\}_{j=1}^k, J_\pm)$ as a data distribution on $\mathbb{R}^d \times \{-1, 1\}$, where each data point (x, y) is generated as follows:

1. *Draw a cluster index $j \sim \mathrm{Unif}([k])$;*
2. *Set $y = +1$ if $j \in J_+$; otherwise $j \in J_-$ and set $y = -1$;*
3. *Draw $x := \mu_j + \xi$, where $\xi \sim \mathcal{N}(0, I_d)$.*

For convenience, we write D instead of $D(\{\mu_j\}_{j=1}^k, J_\pm)$ when these are clear from context. For $s \in \{\pm 1\}$, we write J_s to denote J_+ if $s = +1$ and J_- if $s = -1$.

- Example: $j \sim \mathrm{Unif}([k]) \;\rightarrow\; y = \mathrm{sign}(j \in J_+) \;\rightarrow\; x = \mu_j + \xi$, where $\xi \sim \mathcal{N}(0, I_d)$.

Problem Setup

Input & Output Data Dsitrubtion

Assumption 1 (Orthogonal Equinorm Cluster Features)

The cluster features $\{\mu_j\}_{j=1}^k$ *satisfy* $\|\mu_j\| = \sqrt{d}$ *for all* $j \in [k]$, *and* $\mu_i \perp \mu_j$ *for all* $i \neq j$.

Assumption 2 (Nearly Balanced Classification)

The partition $J_\pm$ *satisfies* $c^{-1} \leq \frac{|J_+|}{|J_-|} \leq c$ *for some constant* $c \geq 1$.

- These assumptions simplify the geometry: orthogonality isolates clusters in feature space, and near balance prevents trivial labeling bias, enabling clean theoretical analysis of optimization and generalization.

Problem Setup

Input & Output Data Dsitrubtion

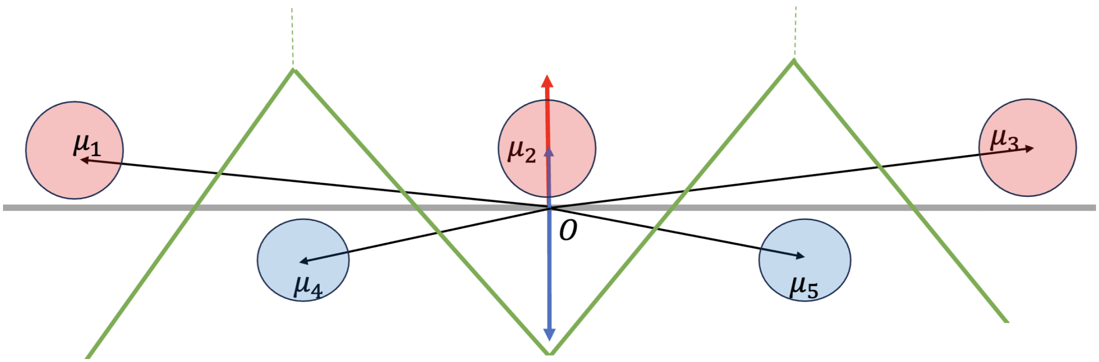


Figure. Cluster-based data distribution used in our analysis.

Definition 2.2 (Two-Layer Neural Network Learner)

We consider a two-layer ReLU network

$$f_\theta(x) = \sum_{j=1}^{2m} a_j \, \sigma(\langle w_j, x \rangle + b_j), \qquad \sigma(z) := \max\{0, z\},$$

where only the first-layer parameters (w_j, b_j) are trainable.
The second-layer weights are fixed and satisfy

$$a_j = \frac{1}{m} \text{ for } j \leq m, \qquad a_j = -\frac{1}{m} \text{ for } j > m.$$

- First m neurons correspond to the positive group and the remaining m neurons to the negative group.

Definition 2.3 (Training Objective and Gradient Descent)

Given training data $\{(x_i, y_i)\}_{i=1}^n$ drawn from D, the empirical loss is

$$\mathcal{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i f_\theta(x_i))\,, \qquad \ell(z) := \log(1 + \exp(-z)).$$

We train the network by gradient descent:

$$\theta^{(t+1)} = \theta^{(t)} - \eta\, \nabla_\theta \mathcal{L}(\theta^{(t)}),$$

where $\eta > 0$ is the step size.

Definition 2.4 (Neuron Activation Indicator)

For each sample x_i and neuron j, define the activation indicator

$$S_{i,j}^{(t)} := \mathbf{1}\left(\langle w_j^{(t)}, x_i\rangle + b_j^{(t)} \geq 0\right).$$

Definition 2.5 (Clean Accuracy)

For a given data distribution D over $\mathbb{R}^d \times \{-1, 1\}$, the clean accuracy of a neural network $f_\theta : \mathbb{R}^d \to \mathbb{R}$ on D is defined as

$$\mathrm{Acc}_D^{\mathrm{clean}}(f_\theta) := \mathbb{P}_{(x,y)\sim D}\big[\,\mathrm{sgn}(f_\theta(x)) = y\big].$$

Definition 2.6 (Robust Accuracy.)

In this work, we focus on the ℓ_2-robustness. The ℓ_2 δ-robust accuracy of f_θ on D is defined as

$$\mathrm{Acc}_D^{\mathrm{robust}}(f_\theta; \delta) := \mathbb{P}_{(x,y)\sim D}\big[\forall \rho \in B_\delta : \mathrm{sgn}(f_\theta(x + \rho)) = y\big],$$

where $B_\delta := \{\rho \in \mathbb{R}^d : \|\rho\| \leq \delta\}$ is the ℓ_2-ball of radius δ.

A network f_θ is said to be δ-robust if

$$\mathrm{Acc}_D^{\mathrm{robust}}(f_\theta; \delta) \geq 1 - \varepsilon(d)$$

for some function $\varepsilon(d) \to 0$ as $d \to \infty$.

Problem Setup

What we have so far

- We have fully specified the learning problem:
  - A structured multi-cluster data distribution.
  - A two-layer ReLU neural network with fixed second-layer signs.
- These components define both the geometry of the data and the dynamics of the learner.
- **Next:** Using this setup, we analyze how gradient descent behaves and what representations the network learns.

Part II

Network Learner Provably Learns Feature-Averaging Solution

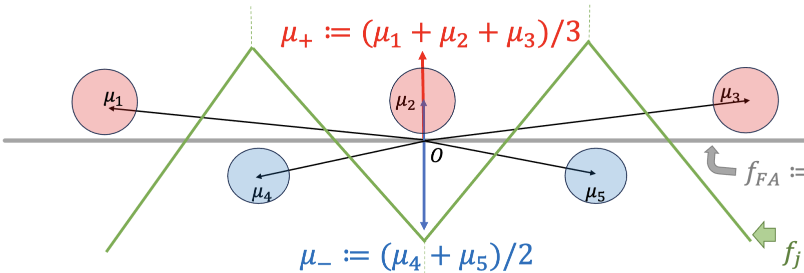
Feature Averaging Network

Feature Averaging vs Feature Decoupling

- **Two possible representation behaviors:**
  - **Feature Averaging:** The network collapses multiple clusters of the same label into a single averaged feature direction.
  - **Feature Decoupling:** The network learns separate directions for individual clusters, preserving fine-grained structure.
- In our setting, gradient descent on the binary-labeled model naturally leads to *feature averaging*.
- This behavior is simple but can severely limit robustness— which we will formalize on the next slides.

Feature Averaging Network

Feature Averaging vs Feature Decoupling



$J_+ = \{1,2,3\}, J_- = \{4,5\}, k = 5,$
$\mu_+ \coloneqq (\mu_1 + \mu_2 + \mu_3)/3,$
$\mu_- \coloneqq (\mu_4 + \mu_5)/2$

$f_{FA} \coloneqq 3ReLU(\langle \mu_+, x \rangle) - 2ReLU(\langle \mu_-, x \rangle)$

$f_{FD} := f_1 + f_2 + f_3 - f_4 - f_5,$
$f_j(x) := ReLU(\langle \mu_j, x \rangle + b_j), j \in [5]$

Definition 1.1 (Feature-Averaging Network)

We define $f_{\mathrm{FA}}(x)$ as the following function:

$$f_{\mathrm{FA}}(x) := |J_+| \cdot \mathrm{ReLU}\left(\langle \mu_+, x \rangle\right) - |J_-| \cdot \mathrm{ReLU}\left(\langle \mu_-, x \rangle\right),$$

where $\mu_+ := \frac{1}{|J_+|}\sum_{j \in J_+} \mu_j$ is the average of cluster centers in the positive class, and similarly $\mu_- := \frac{1}{|J_-|}\sum_{j \in J_-} \mu_j$ is that for the negative class.
We say that a two-layer ReLU network $f_\theta(x)$ is a feature-averaging network iff $f_\theta(x) = C \cdot f_{\mathrm{FA}}(x)$ for some $C > 0$.

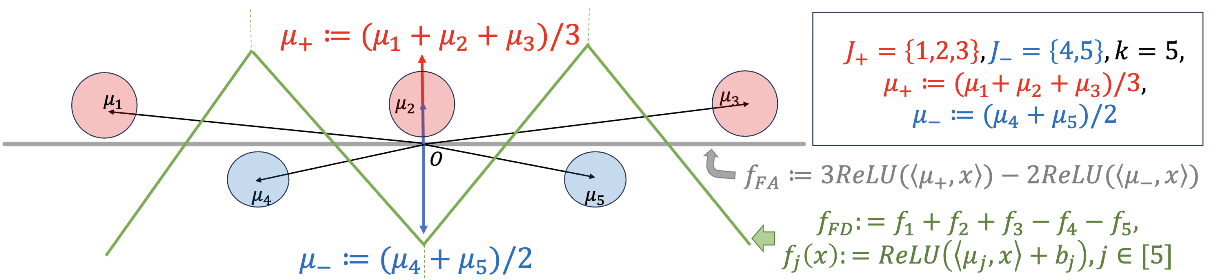
Feature Averaging Network

Feature Averaging Network: Definition

Remark: The feature-averaging network fails to robustly classify perturbed data for a radius larger than $\Omega(\sqrt{d/k})$: in particular, consider the attack vector ρ that aligns with the negative direction of the averaged features, i.e.,

$$\rho \propto -\sum_{j \in J_+} \mu_j + \sum_{j \in J_-} \mu_j.$$

One can easily check that with $\|\rho\| = \delta = \Omega(\sqrt{d/k})$, the attack is successful, i.e., $\operatorname{sgn}(f_{\mathrm{FA}}(x+\rho)) = \operatorname{sgn}(f_{\mathrm{FA}}(x))$ due to the linearity of $f_{\mathrm{FA}}(x+\rho)$ over ρ.



Feature Averaging Network

Why Does Feature Averaging Reduce Robustness?

- In the positive class, we have multiple cluster features μ_j, which are orthogonal and satisfy $\|\mu_j\| = \sqrt{d}$.
- The Feature Averaging Network ignores the individual clusters and uses only the averaged feature

$$\mu_+ = \frac{1}{|J_+|} \sum_{j \in J_+} \mu_j.$$

- Because the μ_j's are orthogonal, averaging spreads out their energy:

$$\|\mu_+\|^2 = \frac{1}{|J_+|^2} \sum_{j \in J_+} \|\mu_j\|^2 = \frac{d}{|J_+|} \quad \Rightarrow \quad \|\mu_+\| = \sqrt{\frac{d}{|J_+|}}.$$

- Thus, each original cluster feature has magnitude $\sqrt{d}$, but their average has only $\sqrt{d/k}$ magnitude.
- A smaller margin means that a perturbation of size $\|\rho\| \approx \sqrt{d/k}$ can flip the prediction, since

$$f(x + \rho) \approx f(x) + \langle \mu_+, \rho \rangle.$$

Assumption 3 (Choices of Hyper-Parameters)

We assume that:

$$d = \Omega(k^{10}) \quad c = \Theta(1) \quad n \in [\Omega(k^7), \exp(O(\log^2(d)))]$$
$$m = \Theta(k) \quad \eta = O(d^{-2}) \quad \sigma_b^2 = \sigma_w^2 = O(\eta k^{-5}).$$

Discussion. We choose these hyper-parameters to place the network in the feature-learning regime:

- (i) **Data Dimension** d : d must be much larger than k so cluster features are orthogonal;
- (ii) **Sufficient Smaples** n : n must be large enough to observe every cluster;
- (iii) **Number of hidden neurons** m : width $m = \Theta(k)$ ensures the model can represent k clusters;
- (iv) small η and initialization prevent activation flips and keep the dynamics stable.

Theorem 1 (Main Claim)

In the setting of training a two-layer ReLU network on the binary classification problem for some $\gamma = o(1)$, after $\Omega(\eta^{-1}) \leq T \leq \exp(\widetilde{O}(k^{1/2}))$ iterations, with probability at least $1-\gamma$, the neural network satisfies the following properties:

1. ***Gradient descent leads the network to the feature-averaging regime:*** *there exists a time-variant coefficient $\lambda(T) \in [\Omega(1), +\infty)$ such that for all $s \in \{\pm 1\}$ and $r \in [m]$, the weight vector satisfies*

$$w_{s,r}^{(T)} - \lambda(T) \sum_{j \in J_s} \|\mu_j\|^{-2} \mu_j \ \leq \ o(d^{-1/2}),$$

   and the bias terms are sufficiently small, i.e., $b_{s,r}^{(T)} \leq o(1)$.

2. ***The clean accuracy is nearly perfect:***

$$\mathrm{Acc}_D^{\mathrm{clean}}(f_{\theta(T)}) \geq 1 - \exp(-\Omega(\log^2 d)).$$

3. ***Consequently, the network is non-robust:*** *for perturbation radius $\delta = \Omega(\sqrt{d/k})$, the δ-robust accuracy is nearly zero:*

$$\mathrm{Acc}_D^{\mathrm{robust}}(f_{\theta(T)}; \delta) \leq \exp(-\Omega(\log^2 d)).$$

GD makes Feature Averaging Network

Preliminaries: weight decomposition

Lemma 1 (Weight Decomposition)

During the training dynamics, there exists the following coefficient sequences $\lambda_{s,r,j}^{(t)}$ and $\sigma_{s,r,i}^{(t)}$ for each $s \in \{-1,+1\}$, $r \in [m]$, $j \in J$, $i \in I$ such that

$$w_{s,r}^{(t)} = w_{s,r}^{(0)} + \sum_{j \in J} \lambda_{s,r,j}^{(t)} \frac{\mu_j}{\|\mu_j\|^2} + \sum_{i \in I} \sigma_{s,r,i}^{(t)} \frac{\xi_i}{\|\xi_i\|^2}.$$

Lemma 2 (Updates of Coefficients)

For each $s \in \{-1,+1\}$, $r \in [m]$, $j \in [k]$, $i \in [n]$ and time $t \geq 0$, the following update equations hold:

$$\lambda_{s,r,j}^{(t+1)} = \lambda_{s,r,j}^{(t)} - \frac{s\eta}{nm} \sum_{i \in I_j} \frac{\ell_i'(t)}{\|\mu_j\|^2} \mathbf{1}\left[r \in S_{s,i}^{(t)}\right],$$

$$\sigma_{s,r,i}^{(t+1)} = \sigma_{s,r,i}^{(t)} - \frac{s\eta}{nm} \frac{\ell_i'(t)}{\|\xi_i\|^2} \mathbf{1}\left[r \in S_{s,i}^{(t)}\right],$$

where $\ell_i'(t) := \ell'(y_i f_{\theta^{(t)}}(x_i))$ denotes the point-wise loss derivative and $I_j := \{i : x_i$ lies in cluster $j\}$.

Proof of Lemma 1.

By the update equation in gradient descent, we know that

$$w_{s,r}^{(t+1)} = w_{s,r}^{(t)} - \frac{s\eta}{nm} \sum_{i \in I} \ell_i'(t)\, x_i\, \mathbf{1}\left[\langle w_{s,r}^{(t)}, x_i \rangle + b_{s,r}^{(t)} \geq 0\right].$$

First, we construct a set of $\{\hat{\lambda}_{s,r,j}^{(t)}\}$ and $\{\hat{\sigma}_{s,r,i}^{(t)}\}$ according to the following recursive formulas:

$$\hat{\lambda}_{s,r,j}^{(t+1)} = \hat{\lambda}_{s,r,j}^{(t)} - \frac{s\eta}{nm} \cdot \sum_{i \in I_j} \frac{\ell_i'(t)}{\|\mu_j\|^2} \mathbf{1}\left[\langle w_{s,r}^{(t)}, x_i \rangle + b_{s,r}^{(t)} \geq 0\right], \quad (1)$$

$$\hat{\sigma}_{s,r,i}^{(t+1)} = \hat{\sigma}_{s,r,i}^{(t)} - \frac{s\eta}{nm} \cdot \frac{\ell_i'(t)}{\|\xi_i\|^2} \mathbf{1}\left[\langle w_{s,r}^{(t)}, x_i \rangle + b_{s,r}^{(t)} \geq 0\right], \quad (2)$$

with initialization

$$\hat{\lambda}_{s,r,j}^{(0)} = 0, \qquad \hat{\sigma}_{s,r,i}^{(0)} = 0. \quad (3)$$

Now, we prove by induction on t that $\{\hat{\lambda}_{s,r,j}^{(t)}\}$ and $\{\hat{\sigma}_{s,r,i}^{(t)}\}$ constructed as above satisfy that

$$w_{s,r}^{(t)} = w_{s,r}^{(0)} + \sum_{j \in J} \hat{\lambda}_{s,r,j}^{(t)} \frac{\mu_j}{\|\mu_j\|^2} + \sum_{i \in I} \hat{\sigma}_{s,r,i}^{(t)} \frac{\xi_i}{\|\xi_i\|^2}. \quad (4)$$

Proof of Lemma 1.

$$w_{s,r}^{(t+1)} = w_{s,r}^{(t)} - \frac{s\eta}{nm} \sum_{i \in I} \ell_i'(t)\, x_i \mathbf{1}\left[\langle w_{s,r}^{(t)}, x_i \rangle + b_{s,r}^{(t)} \geq 0\right] \tag{5}$$

$$= w_{s,r}^{(t)} - \frac{s\eta}{nm} \sum_{i \in I} \ell_i'(t)\, (\mu_{c(i)} + \xi_i) \mathbf{1}\left[\langle w_{s,r}^{(t)}, x_i \rangle + b_{s,r}^{(t)} \geq 0\right] \tag{6}$$

$$= w_{s,r}^{(t)} - \frac{s\eta}{nm} \left(\sum_{j \in J} \mu_j \sum_{i \in I_j} \ell_i'(t)\, \mathbf{1}\left[\langle w_{s,r}^{(t)}, x_i \rangle + b_{s,r}^{(t)} \geq 0\right] + \sum_{i \in I} \xi_i \ell_i'(t)\, \mathbf{1}\left[\langle w_{s,r}^{(t)}, x_i \rangle + b_{s,r}^{(t)} \geq 0\right] \right). \tag{7}$$

Hence,

$$w_{s,r}^{(t+1)} = w_{s,r}^{(0)} + \sum_{j \in J} \frac{\mu_j}{\|\mu_j\|^2} \left(\hat{\lambda}_{s,r,j}^{(t)} - \frac{s\eta}{nm} \sum_{i \in I_j} \frac{\ell_i'(t)}{\|\mu_j\|^2} \mathbf{1}\left[\langle w_{s,r}^{(t)}, x_i \rangle + b_{s,r}^{(t)} \geq 0\right] \right) \tag{8}$$

$$+ \sum_{i \in I} \frac{\xi_i}{\|\xi_i\|^2} \left(\hat{\sigma}_{s,r,i}^{(t)} - \frac{s\eta}{nm} \frac{\ell_i'(t)}{\|\xi_i\|^2} \mathbf{1}\left[\langle w_{s,r}^{(t)}, x_i \rangle + b_{s,r}^{(t)} \geq 0\right] \right) \tag{9}$$

$$= w_{s,r}^{(0)} + \sum_{j \in J} \hat{\lambda}_{s,r,j}^{(t+1)} \frac{\mu_j}{\|\mu_j\|^2} + \sum_{i \in I} \hat{\sigma}_{s,r,i}^{(t+1)} \frac{\xi_i}{\|\xi_i\|^2}. \tag{10}$$

GD makes Feature Averaging Network

Proof Step 1: Activation Stability

A key step in the dynamics is that the activation patterns of neurons remain stable during early training.

Lemma 3 (Activation Stability)

For all $t \le T_0$, with high probability,

$$S^{(t)}_{+1,i} = [m], \qquad \forall i \in I_+, \qquad \textit{and} \qquad S^{(t)}_{-1,i} = [m], \qquad \forall i \in I_-.$$

Expanded form of the activation sets:

$$S^{(t)}_{s,i} = \left\{ r \in [m] \;:\; \underbrace{\langle w^{(t)}_{s,r}, x_i \rangle + b^{(t)}_{s,r}}_{\text{ReLU pre-activation}} \ge 0 \right\}.$$

Thus the lemma above states that:

$$\forall i \in I_+,\ \forall r \in [m],\ \langle w^{(t)}_{+1,r}, x_i \rangle + b^{(t)}_{+1,r} \ge 0 \qquad \text{and} \qquad \forall i \in I_-,\ \forall r \in [m],\ \langle w^{(t)}_{-1,r}, x_i \rangle + b^{(t)}_{-1,r} \ge 0.$$

- Therefore, every positive neuron activates on every positive sample, and similarly for the negative.
- Consequently, the indicator in Lemma 4 becomes

  $$\mathbf{1}[r \in S^{(t)}_{s,i}] = 1 \quad \text{whenever } y_i = s,$$

  which greatly simplifies the coefficient dynamics.

GD makes Feature Averaging Network

Proof Step 1: Activation Stability

A key step in the dynamics is that the activation patterns of neurons remain stable during early training.

Lemma 4 (Updates of Coefficients (changed by Lemma 3))

For each $s \in \{-1, +1\}$, $r \in [m]$, $j \in [k]$, $i \in [n]$ and time $t \geq 0$, the following update equations hold:

$$\lambda_{s,r,j}^{(t+1)} = \lambda_{s,r,j}^{(t)} - \frac{s\eta}{nm} \sum_{i \in I_j} \frac{\ell_i'(t)}{\|\mu_j\|^2},$$

$$\sigma_{s,r,i}^{(t+1)} = \sigma_{s,r,i}^{(t)} - \frac{s\eta}{nm} \frac{\ell_i'(t)}{\|\xi_i\|^2},$$

where $\ell_i'(t) := \ell'(y_i f_{\theta^{(t)}}(x_i))$ denotes the point-wise loss derivative and $I_j := \{i : x_i \text{ lies in cluster } j\}$.

GD makes Feature Averaging Network

Proof Step 2: Margin Balance Within Each Label

Lemma 5 (Margin Balance)

For any i, i' with $y_i = y_{i'} = s$, we have

$$\frac{\ell'_i(t)}{\ell'_{i'}(t)} = 1 \pm o(1) \qquad \textit{for all } t \leq T_0.$$

Thus all samples from the same label contribute approximately equally to the coefficient updates. Combined with the balanced cluster sizes $|I_j| \approx n/k$, this yields

$$\sum_{i \in I_j} \ell'_i(t) = \frac{n}{k}\,\ell'_s(t)(1 \pm o(1)) \qquad \text{for all } j \in J_s.$$

GD makes Feature Averaging Network

Proof Step 3: Emergence of Feature Averaging

A more refined derivation of why the coefficients $\lambda_{s,r,j}^{(t)}$ become nearly identical for all clusters $j \in J_s$.

(A) Nearly identical updates within a label class.

Since each cluster $j \in J_s$ contains approximately n/k samples (by Assumption 2), we obtain

$$\sum_{i \in I_j} \ell_i'(t) = \frac{n}{k} \ell_s'(t)(1 \pm o(1)).$$

Substituting this into the update rule of Lemma 4 gives

$$\lambda_{s,r,j}^{(t+1)} = \lambda_{s,r,j}^{(t)} - \frac{s\eta}{nm} \sum_{i \in I_j} \frac{\ell_i'(t)}{\|\mu_j\|^2} \qquad \forall j \in J_s.$$

$$\lambda_{s,r,j}^{(t+1)} = \lambda_{s,r,j}^{(t)} - \frac{s\eta}{k\|\mu_j\|^2} \ell_s'(t)(1 \pm o(1)), \qquad \forall j \in J_s.$$

Thus the increments $\lambda_{s,r,j}^{(t+1)} - \lambda_{s,r,j}^{(t)}$ are uniform across all clusters in J_s.

(B) Convergence of coefficients to a common value.

Since all clusters in J_s receive asymptotically identical updates, the difference between any pair of coefficients evolves as

$$\lambda_{s,r,j}^{(t+1)} - \lambda_{s,r,j'}^{(t+1)} = \left(\lambda_{s,r,j}^{(t)} + \Delta\lambda_j^{(t)}\right) - \left(\lambda_{s,r,j'}^{(t)} + \Delta\lambda_{j'}^{(t)}\right).$$

By Step 3(A), the updates satisfy

$$\Delta\lambda_j^{(t)} = \Delta\lambda_{j'}^{(t)}(1 \pm o(1)), \qquad \Delta\lambda_j^{(t)} - \Delta\lambda_{j'}^{(t)} = o(1),$$

hence

$$\lambda_{s,r,j}^{(t+1)} - \lambda_{s,r,j'}^{(t+1)} = \left(\lambda_{s,r,j}^{(t)} - \lambda_{s,r,j'}^{(t)}\right)(1 \pm o(1)).$$

Iterating over $t = 0, 1, \ldots, T_0$ shows that these differences contract to $o(1)$, which implies

$$\lambda_{s,r,j}^{(T)} = \lambda_s(T)\|\mu_j\|^{-2} \pm o(1), \qquad \forall j \in J_s,$$

for a common scalar $\lambda_s(T) \in [\Omega(1), \infty)$.

(C) Substitution into the weight decomposition.

Plugging the above estimate into the decomposition of Lemma 1 and using the fact that the noise coefficients contribute only $o(d^{-1/2})$, we obtain

$$w_{s,r}^{(T)} = \lambda_s(T) \sum_{j \in J_s} \|\mu_j\|^{-2} \mu_j + o(d^{-1/2}).$$

Hence every neuron in label group s aligns with the averaged cluster feature

$$\mu_s^{\mathrm{avg}} = \sum_{j \in J_s} \|\mu_j\|^{-2} \mu_j.$$

This establishes the emergence of the *feature-averaging regime*.

Clean Accuracy of the Feature-Averaging Network

Theorem 2

There exist values of b_+ and b_- such that the feature-averaging network $f_{\theta_{\mathrm{avg}}}$ achieves $1 - o(1)$ standard accuracy over D.

Proof.

Recall that in the feature-averaging regime,

$$w_+ = \sum_{j \in J_+} \mu_j, \qquad w_- = \sum_{j \in J_-} \mu_j, \qquad f_{\theta_{\mathrm{avg}}}(x) = \sum_{j \in J_+} \langle \mu_j, x \rangle - \sum_{j \in J_-} \langle \mu_j, x \rangle.$$

Now substitute the sample representation $x = \alpha \mu_i + \xi$ for a point in positive cluster i, and choose $b_+ = b_- = 0$. Then, with high probability,

$$f_{\theta_{\mathrm{avg}}}(x) \geq \langle \mu_i, \alpha \mu_i \rangle + \sum_{j \in J_+ \setminus \{i\}} \langle \mu_j, \xi \rangle - \sum_{j \in J_-} \langle \mu_j, \xi \rangle.$$

Since $\quad \langle \mu_i, \alpha \mu_i \rangle = \Theta(d), \qquad |\langle \mu_j, \xi \rangle| \leq O(\Delta) = O\Big(\sigma \sqrt{d} \ln d\Big),$

we obtain $\quad f_{\theta_{\mathrm{avg}}}(x) \geq \Theta(d) - O(k\Delta) = \Theta(d) - O\Big(k\sigma \sqrt{d} \ln d\Big) \geq 0.$

Thus $f_{\theta_{\mathrm{avg}}}$ correctly classifies (x, y) with high probability. $\square$

GD makes Feature Averaging Network

Clean Accuracy and Robust Accuracy of the Feature-Averaging Network

Theorem 3

Consequently, the network is non-robust. For a perturbation radius $\delta = \Omega\left(\frac{pd}{k}\right)$, *the* δ*-robust accuracy satisfies* $\mathrm{Acc}_{\mathcal{D}}^{\mathrm{robust}}\left(f_\theta(T), \delta\right) \le \exp\left(-\Omega(\log^2 d)\right)$.

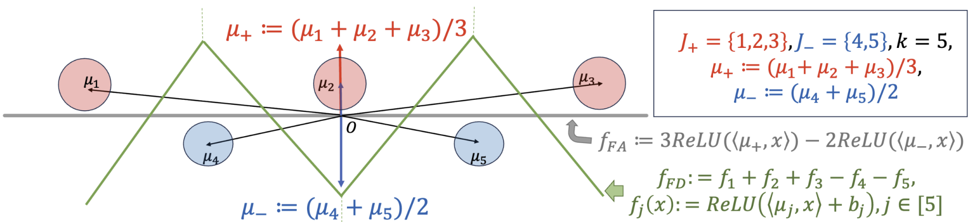
Feature Averaging Network

Feature Averaging Network: Definition

Remark: The feature-averaging network fails to robustly classify perturbed data for a radius larger than $\Omega(\sqrt{d/k})$: in particular, consider the attack vector ρ that aligns with the negative direction of the averaged features, i.e.,

$$\rho \propto -\sum_{j \in J_+} \mu_j + \sum_{j \in J_-} \mu_j.$$

One can easily check that with $\|\rho\| = \delta = \Omega(\sqrt{d/k})$, the attack is successful, i.e., $\mathrm{sgn}(f_{\mathrm{FA}}(x+\rho)) = \mathrm{sgn}(f_{\mathrm{FA}}(x))$ due to the linearity of $f_{\mathrm{FA}}(x+\rho)$ over ρ.



Part III

Fine-Grained supervision improves robustness

Feature Decoupling Network

Set Up

- **Training Set**
  First, Sample a training set $\mathcal{S} := \{(x_i, y_i)\}_{i=1}^{n} \subseteq \mathbb{R}^d \times \{\pm 1\}$ from $\mathcal{D}$,
  along with the cluster labels $\{\tilde{y}_i\}_{i=1}^{n}$ for all data points.
  Then a k-class neural network classifier is trained on $\tilde{\mathcal{S}} := \{(x_i, \tilde{y}_i)\}_{i=1}^{n} \subseteq \mathbb{R}^d \times [k]$.
- **Multi-Class Network Classifier**
  Train the following two-layer neural network: $\boldsymbol{F}_{\boldsymbol{\theta}}(x) := (f_1(x), f_2(x), \ldots, f_k(x)) \in \mathbb{R}^k$,
  where $f_j(x) := \frac{1}{h}\sum_{r=1}^{h} \mathrm{ReLU}(\langle \boldsymbol{w}_{j,r}, \boldsymbol{x}\rangle)$, $\theta := (w_{1,1}, w_{1,2}, \ldots, w_{k,h}) \in \mathbb{R}^{khd}$,
  $h = \Theta(1)$ is the width of each sub-network.
  $\boldsymbol{F}_{\boldsymbol{\theta}}(x)$ are converted to probabilities, namely $p_j(x) := \frac{\exp(f_j(x))}{\sum_{i=1}^{k}\exp(f_i(x))}$ for $j \in [k]$.
  For predicting the binary label, $\boldsymbol{F}_{\theta}(x) := \sum_{j\in J_+} p_j(x) - \sum_{j\in J_-} p_j(x)$.
- **Training Objective**
  Using cross-entropy loss : $\mathcal{L}_{CE}(\theta) := -\frac{1}{n}\sum_{i=1}^{n} \log p_{\tilde{y}_i}(x_i)$
  Gradient descent: $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}_{CE}(\boldsymbol{F}_{\boldsymbol{\theta}^{(t)}})$
  At initialization, $w_{j,r}^{(0)} \sim \mathcal{N}(0, \sigma_w^2 I_d)$ for some $\sigma_w > 0$.

Theorem 4 (Main Claim)

In the setting of training a two-layer ReLU network on the multiple classification problem for some $\gamma = o(1)$, after $\Omega(\eta^{-1}k^8) \le T \le \exp(\widetilde{O}(k^{1/2}))$ iterations, with probability at least $1-\gamma$, the neural network satisfies the following properties:

1. ***The network converges to the feature-decoupling regime:***
   The network converges to the feature-decoupling regime: there exists a time-variant coefficient $\lambda^{(T)} \in [\Omega(\log k), +\infty)$ such that for all $j \in [k]$, $r \in [h]$, the weight vector $\mathbf{w}_{j,r}^{(T)}$ can be approximated as

$$\left\| w_{j,r}^{(T)} - \lambda^{(T)} \|\boldsymbol{\mu}_j\|^{-2} \boldsymbol{\mu}_j \right\| \le o(d^{-1/2}).$$

2. ***The clean accuracy is nearly perfect:***

$$\mathrm{Acc}_{clean}^{\mathcal{D}}(F_{\theta^{(T)}}^{binary}) \ge 1 - \exp(-\Omega(\log^2 d)).$$

3. ***The corresponding binary classifier achieves optimal robustness:***
   for perturbation radius $\delta = O(\sqrt{d})$, the δ-robust accuracy is also nearly perfect, i.e., $Acc_{robust}^{\mathcal{D}}(F_{\theta^{(T)}}^{binary}; \delta) \ge 1 - \exp(-\Omega(\log^2 d))$.

Lemma 6

Assuming the inductive hypotheses hold before time step t, for all $r \in [h], s, j \in J$, we have

$$\left| \langle \boldsymbol{w}_{s,r}^{(t)}, \boldsymbol{\mu}_j \rangle - \lambda_{s,r,j}^{(t)} \right| \leq \frac{\epsilon}{6}.$$

Lemma 7

Assuming the inductive hypotheses hold before time step t, for $i \in I, s \in J, s \neq c(i), r \in [m]$, we have

$$|\lambda_{s,r,c(i)}^{(t)}| \leq \epsilon, |\sigma_{s,r,i}^{(t)}| \leq 2\epsilon.$$

Lemma 8

For all $s \in J, r \in [h]$, We have

$$\sqrt{d}\left\|\boldsymbol{w}_{s,r}^{(T)} - \lambda^{(T)}\boldsymbol{\mu}_s\|\boldsymbol{\mu}_s\|^{-2}\right\| = o(1).$$

Lemma 9

Let $\boldsymbol{\xi} \sim \mathcal{N}(0, I_d)$. Then, with probability at least $1 - 2nd^{-\ln(d)/2}$, for all $s \in J, r \in [h]$ we have

$$|\langle \boldsymbol{w}_{s,r}^{(T)}, \boldsymbol{\xi}\rangle| \leq \frac{\epsilon}{6}.$$

Lemma 10

For all $j \in J, r \in [h]$, we have $\frac{\ln(T\eta)}{4} \leq \lambda_{j,r,j}^{(T)} \leq 4\ln(T+1)$.

Lemma 11

$$|\lambda^{(T)} - \lambda_{j,r,j}^{(T)}| \leq 204k^2\epsilon\lambda^{(T)},$$
$$\lambda^{(T)} \leq 4\ln(T+1),$$
$$\lambda^{(T)} \geq \frac{\ln(T\eta)}{4} \geq 2\ln(k) = \Omega(\log(k)).$$

Lemma 12

$\epsilon = max\left\{\frac{2\ln(\frac{n}{k})}{\sqrt{\frac{n}{k}} - \ln(\frac{n}{k})}, \frac{k^2\Delta}{d}, \frac{k^2}{n}\right\}$. $\epsilon = o(k^{-2.5})$ *according to our hyper-parameter settings.*

For items 1 and 2 of Theorem 4, the proof process is identical to that of Feature Averaging.

Theorem 5

For perturbation radius $\delta = O(\sqrt{d})$, the δ-robust accuracy is also nearly perfect

Proof.

By lemma 8, we know that $\sqrt{d}\|\mathbf{w}_{s,r}^{(T)}\| \leq \lambda^{(T)} + o(1) \leq 2\lambda^{(T)}$.

Then for any perturbation $\boldsymbol{\rho}$ with $\boldsymbol{\rho} \leq \frac{\sqrt{d}}{10}$. $(\because \|\mathbf{w}_{s,r}^{(T)}\|\|\boldsymbol{\rho}\| \leq \frac{\lambda^{(T)}}{5})$

We know that

$$\begin{aligned}
\langle \mathbf{w}_{j,r}^{(T)}, \mathbf{x} + \boldsymbol{\rho} \rangle &= \langle \mathbf{w}_{j,r}^{(T)}, \boldsymbol{\mu_i} \rangle + \langle \mathbf{w}_{j,r}^{(T)}, \boldsymbol{\xi} \rangle + \langle \mathbf{w}_{j,r}^{(T)}, \boldsymbol{\rho} \rangle \\
&\geq \lambda_{j,r,j}^{(T)} - \frac{\epsilon}{6} - \frac{\epsilon}{6} - \|\mathbf{w}_{j,r}^{(T)}\|\|\boldsymbol{\rho}\| \ (\because \text{Lemma 6,9}) \\
&\geq \frac{3\lambda^{(T)}}{4} \ (\because \text{Lemma 10,11,12})
\end{aligned}$$

$\square$

Proof.

For $s \in J, s \neq j$, we know that

$$\begin{aligned}
\langle \boldsymbol{w}_{s,r}^{(T)}, \boldsymbol{x} + \boldsymbol{\rho} \rangle &= \langle \boldsymbol{w}_{s,r}^{(T)}, \boldsymbol{\mu_i} \rangle + \langle \boldsymbol{w}_{s,r}^{(T)}, \boldsymbol{\xi} \rangle + \langle \boldsymbol{w}_{s,r}^{(T)}, \boldsymbol{\rho} \rangle \\
&\leq \lambda_{s,r,j}^{(T)} + \frac{\epsilon}{6} + \frac{\epsilon}{6} + \|\boldsymbol{w}_{s,r}^{(T)}\| \|\boldsymbol{\rho}\| \ (\because \text{Lemma 6,9}) \\
&\leq \epsilon + \frac{\epsilon}{3} + \frac{\lambda^{(T)}}{5} \ (\because \text{Lemma 7}) \\
&\leq \frac{3\lambda^{(T)}}{4} - \ln(k) \ (\because \text{Lemma 10,11,12})
\end{aligned}$$

Thus we know that $f_j(\boldsymbol{x} + \boldsymbol{\rho}) \geq \dfrac{3\lambda^{(T)}}{4}$ and $f_s(\boldsymbol{x} + \boldsymbol{\rho}) \leq \dfrac{3\lambda^{(T)}}{4} - \ln(k)$.

$\square$

Proof.

let $G(\boldsymbol{x})$ denote the numerator of $F_{\boldsymbol{\theta}^{(T)}}^{binary}(\boldsymbol{x})$, where denominator is $\sum_{s \in J} e^{f_s(\boldsymbol{x})}$. We know

$$\mathrm{sgn}(F_{\boldsymbol{\theta}^{(T)}}^{binary}) = \mathrm{sgn}(G).$$

Let us first consider the case where $G(\boldsymbol{x}) \geq 0$.
Thus we have

$$\begin{aligned} G(\boldsymbol{x}+\boldsymbol{\rho}) &= \sum_{j \in J_+} \exp\left(f_j(\boldsymbol{x}+\boldsymbol{\rho})\right) - \sum_{j \in J^-} \exp\left(f_j(\boldsymbol{x}+\boldsymbol{\rho})\right) \\ &\geq \exp\left(3\lambda^{(T)}/4\right) - \sum_{j \in J_-} \exp\left(3\lambda^{(T)}/4 - \ln(k)\right) \\ &\geq 0. \end{aligned}$$

That is to say $\mathrm{sgn}(G(\boldsymbol{x}+\boldsymbol{\rho})) = \mathrm{sgn}(G(\boldsymbol{x}))$,

which means $F_{\boldsymbol{\theta}^{(T)}}^{binary}$ is robust under any perturbation with radius smaller than $\dfrac{\sqrt{d}}{10}$. □

Part IV

EXPERIMENTS

Experiments
Dataset

- **Synthetic Dataset**
  - Generate synthetic data following the approach described in Input & Output Data Distribution.
  - Hyperparameters:
    $k = 10,\ d = 3072,\ m = 5,\ n = 1000,\ \alpha = \sigma = 1,\ \eta = 0.001,\ \sigma_w = \sigma_b = 10^{-5},\ T = 100.$
  - The first 5 clusters are labeled as positive and the remaining 5 as negative.
- **CIFAR-10**
  - Merge the first 5 classes into one (positive) class.
  - Merge the remaining 5 classes into the other (negative) class.
  - Use the standard **10-class CIFAR-10 classification** as the multi-class (10-way) task for comparison.

Experiments

Method: The method of constructing neural networks

- Use a ResNet18 model pre-trained on CIFAR-10.
- Replace the original final layer with a two-layer ReLU network:

$$f_j(z) = \frac{1}{h}\sum_{r=1}^{h} \mathrm{ReLU}(\langle w_{j,r}, z\rangle), \quad j \in [10],$$

  where z is the hidden representation of the penultimate layer.
- Only the last two layers of the whole network are trained; the pre-trained backbone is frozen.
- Set the width of the first layer to 30 hidden units ($h = 3$ neurons per class),
  so that the accuracy of the pre-trained model is not compromised.
- For the binary network with 15 positive and 15 negative neurons, we equally divide them into 5 positive and 5 negative classes to ensure a fair comparison, so that both models share the same form

$$F := (f_1, f_2, \ldots, f_{10}) \in \mathbb{R}^{10},$$

  and each sub-network f_j corresponds to a weight vector w_j.

Experiments

Method: Explanation of Symbols Required for Regime Interpretation

- Inspired by the theoretical study of neuron collapse (Papyan et al., 2020), define the class feature

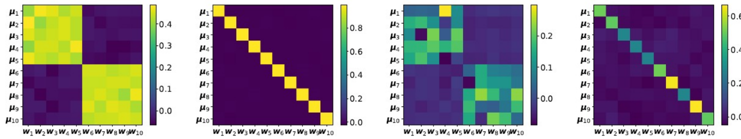
$$\mu_i := \text{average penultimate-layer output of class } i, \quad i \in [10].$$

- For 10-class classification, define the equivalent weight of sub-network f_j as

$$w_j := \frac{1}{h}\sum_{r=1}^{h} w_{j,r},$$

EXPERIMENTS
RESULT

- **Feature-averaging & Feature-decoupling regime**



(a) SynData: 2-class (b) SynData: 10-class (c) CIFAR-10: 2-class (d) CIFAR-10: 10-class

Figure 2: Illustration of feature averaging and feature decoupling on synthetic dataset (a,b) and CIFAR-10 dataset (c,d). Figure (a) and Figure (c) correspond to models trained using 2-class labels, and Figure (b) and Figure (d) correspond to models trained using 10-class labels, respectively. Each element in the matrix, located at position (i, j), represents the average cosine value of the angle between the feature vector $\boldsymbol{\mu}_i$ of the i-th feature and the equivalent weight vector $\boldsymbol{w}_j$ of the $f_j(\cdot)$.

- **Robustness improvement**

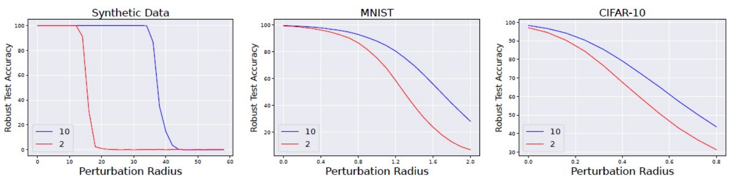


Figure 3: **Verifying robustness improvement:** We compare adversarial robustness between model trained by 2-class labels (red line) and model trained by 10-class labels (blue line) on synthetic data (the left), MNIST (the middle) and CIFAR-10 (the right).

References I