UML Chapter 30: Compression Bounds & Stronger Generalization Bounds for Deep Nets via a Compression Approach

Wangzhi Zhan

June 2024

Outline

- Introduction
- Data Compression
- Model Compression
- Summary

Introduction

Why to care about generalization?

Safety Concern:

- **If a model generalizes well, it can perform more robustly**

Understanding Machine Learning

- **The most important feature of ML compared with a dictionary is generalization**

Estimating Number of Samples Needed

- **Samples can be costly to collect, hence the importance of sample number estimation**

…

Cruise recalls all self-driving cars after grisly accident and California ban

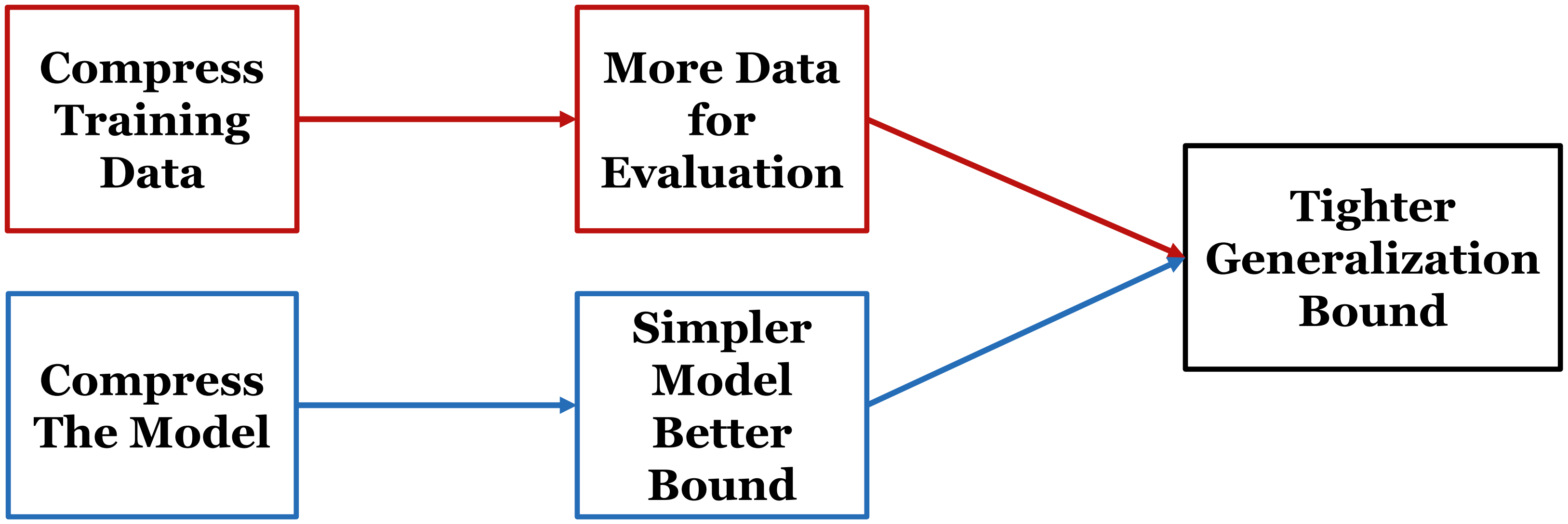
All 950 of the General Motors subsidiary's autonomous cars will be taken off roads for a software update

The image is taken from https://www.theguardian.com/technology/2023/nov/08/cruise-recall-self-driving-cars-gm

Introduction

☐ To better guarantee generalization

- Chapter 30 of UML:
- Compress the training data
- S. Arora et al. 2018
- Compress the model



The image is taken from https://www.theguardian.com/technology/2023/nov/08/cruise-recall-self-driving-cars-gm

Outline

- Introduction
- Data Compression
- Model Compression
- Summary

Data Compression

☐ Motivation

Notations:

- T: Training Set

$\Updownarrow$ Independent

- V: Validation Set
- D: Data Distribution (Unknown)
- h: Hypothesis

$\xRightarrow{\text{Bernstein's Inequality}}$

(Lemma 30.1) With probability $1-\delta$ we have:

$$L_D(h_T) \le L_V(h_T) + \sqrt{\frac{2L_V(h_T)\log(1/\delta)}{|V|}} + \frac{4\log(1/\delta)}{|V|}$$

Questions:

- How to sample T and V?
- How to balance $|T|$ and $|V|$?

$\xRightarrow{\text{Solution}}$

Data Compression

Data Compression

Problem of Independence

If T is selected after looking at the whole set S, then T and V are not independent.

Solution:

- **Choose $I \in [m]^k$ before looking at S**
- **Lemma 30.1 holds**

$$P\left(L_D(h_I) \geq L_V(h_I) + \sqrt{\frac{2L_V(h_I)\log(1/\delta)}{|V|}} + \frac{4\log(1/\delta)}{|V|}\right) \leq \delta$$

- **Number of all possible I is $C_m^k < m^k$**
- **Then we have:**

$$P\left(\exists I, \text{s.t. } L_D(h_I) \geq L_V(h_I) + \sqrt{\frac{2L_V(h_I)\log(1/\delta)}{|V|}} + \frac{4\log(1/\delta)}{|V|}\right) \leq m^k\delta$$

Data Compression

Problem of Independence

- **Then we have:**

$$P\left(\exists I, \mathrm{s.t.}\ L_D(h_I) \geq L_V(h_I) + \sqrt{\frac{2L_V(h_I)\log(1/\delta)}{|V|} + \frac{4\log(1/\delta)}{|V|}}\right) \leq m^k\delta$$

- **Denote $\delta' = m^k\delta$, and I^* as the "best" I, we get (Theorem 30.2) :**

$$P\left(L_D(h_{I^*}) \geq L_V(h_{I^*}) + \sqrt{\frac{4kL_V(h_{I^*})\log(m/\delta')}{m} + \frac{8k\log(m/\delta')}{m}}\right) \leq \delta'$$

- **Equivalent to:**

$$P\left(L_D(h_{I^*}) \leq L_V(h_{I^*}) + \sqrt{\frac{4kL_V(h_{I^*})\log(m/\delta')}{m} + \frac{8k\log(m/\delta')}{m}}\right) \geq 1 - \delta'$$

Data Compression

Problem of Independence

- **Equivalent to:**

$$P\left(L_D(h_{I^*}) \le L_V(h_{I^*}) + \sqrt{\frac{4kL_V(h_{I^*})\log(m/\delta')}{m}} + \frac{8k\log(m/\delta')}{m}\right) \ge 1-\delta'$$

- **Comparing to Lemma 30.1**

$$L_D(h_T) \le L_V(h_T) + \sqrt{\frac{2L_V(h_T)\log(1/\delta)}{|V|}} + \frac{4\log(1/\delta)}{|V|}$$

Remarks:

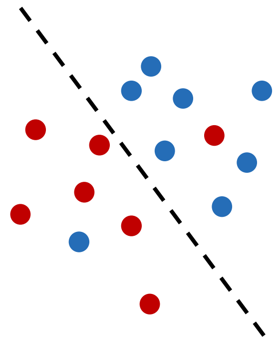
- **The bound is less tight than Lemma 30.1**
- **We can choose I^* to get lower L_V**
- **A small k will be good**

Data Compression

☐ Definitions of Compression Scheme

Notations:

- ***A***: **Sampling Algorithm (Select *k* out of *m* examples)**
- ***B***: **Learning Algorithm (Learn *h* based on k examples)**
- ***H***: **Hypothesis Class**



Definition 1:

- **We say *H* has a compression scheme if for all *h*∈*H*:**

$$\{(x_i, h(x_i))\}_{i=1}^{m} \xrightarrow{\boldsymbol{A}} \{(x_{i_{k'}}, h(x_{i_{k'}}))\}_{k'=1}^{k} \xrightarrow{\boldsymbol{B}} h', \text{s.t.}\, L_S(h') = 0$$

Definition 2:

- **We say *H* has a compression scheme if for all *h*∈*H*:**

$$\{(x_i, y_i)\}_{i=1}^{m} \xrightarrow{\boldsymbol{A}} \{(x_{i_{k'}}, y_{i_{k'}})\}_{k'=1}^{k} \xrightarrow{\boldsymbol{B}} h', \text{s.t.}\, L_S(h') \le L_S(h)$$

Remark:

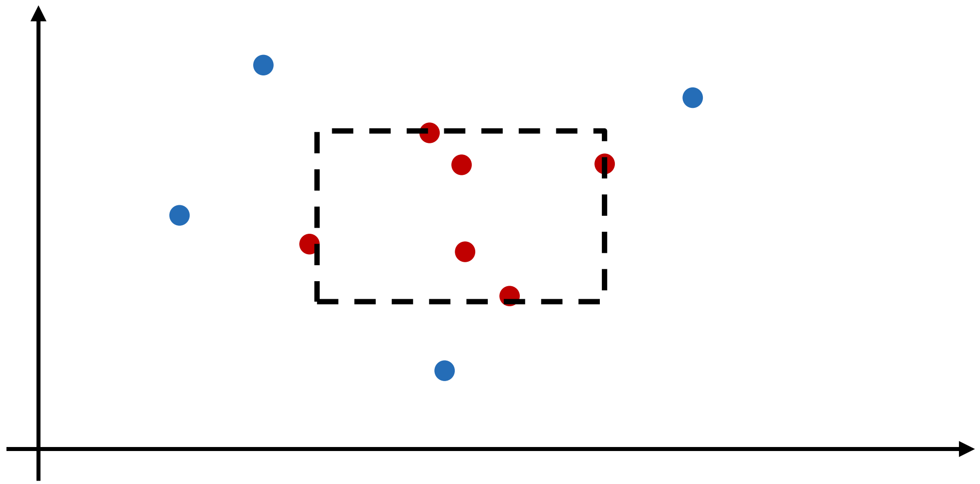
- **Definition 1 uses “cleaner” data**
- **Definition 1 implies Definition 2**

Data Compression

Examples of Compression Scheme

Axis Aligned Rectangles:

- **Use a single rectangle to classify positive points**
- **k = 2d, 2 extreme values on each dimension**

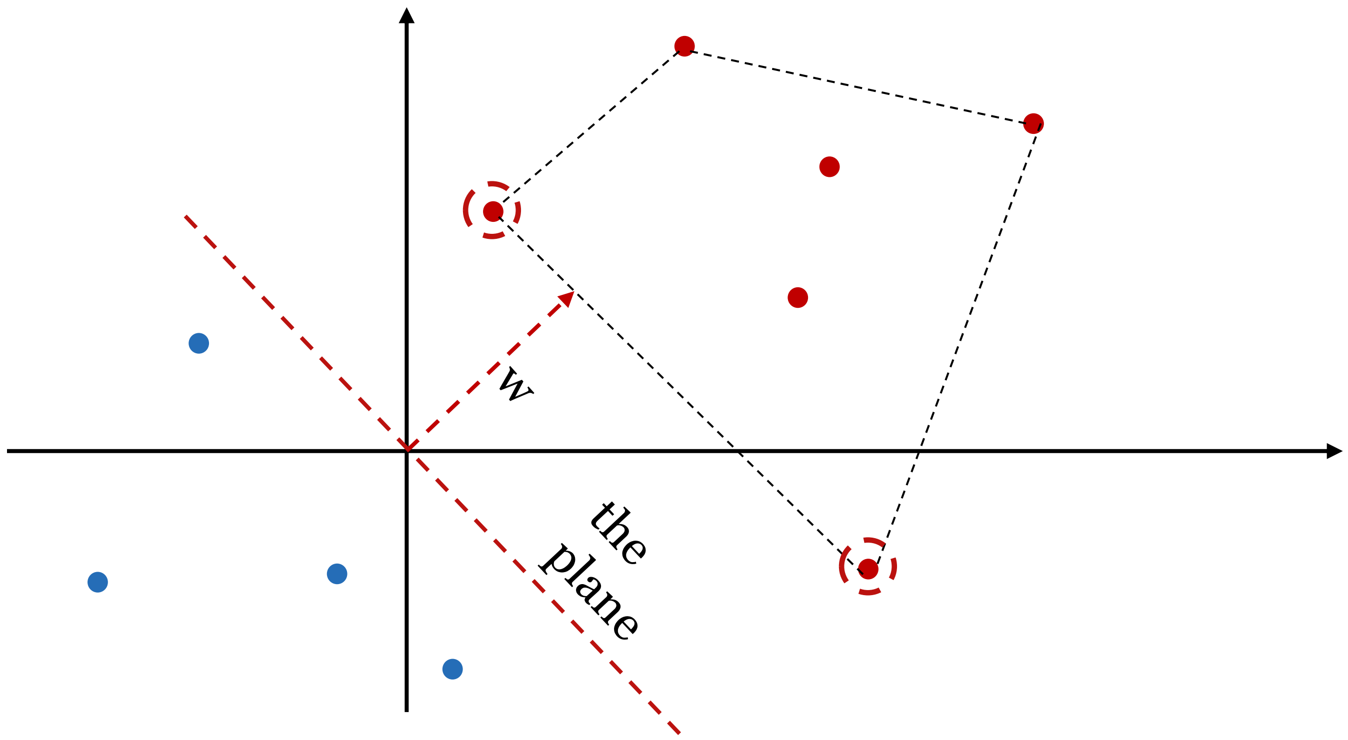


Data Compression

◻ Examples of Compression Scheme

Half Spaces:

- **Use a plane to linear separate positive and negative points**
- **$k = d$, choose the segment of convex hull that is closest to the origin**



Data Compression

Examples of Compression Scheme

Separating Polynomials:

- $f(x) = \text{sign}(p(x))$, p is a polynomial of degree r
- p has $O(d^r)$ terms
- This problem reduces to half spaces of $d' = d^r$
- $k = d^r$

Separation with Margin

- Training set has margin γ
- Perceptron needs at most $1/\gamma^2$ updates to converge
- $k \leq 1/\gamma^2$

Data Compression

Section Summary

On Compression Bounds:

- The reason for data compression: to get tighter generalization bounds
- Derived a bound (Theorem 30.2)

On Compression Schemes:

- Two definitions of compression schemes
- Four practical examples

Outline

- Introduction
- Data Compression
- Model Compression
- Summary

Model Compression

Motivation

problem faced

$\left|L(f) - \hat{L}(f)\right| \leq$ **some bound**

f is complicated

a loose bound

Theorem 2.2. *((Neyshabur et al., 2017a)) For any deep net with layers $A^1, A^2, \ldots A^d$ and output margin γ on a training set S, the generalization error can be bounded by*

$$\tilde{O}\left(\sqrt{\frac{hd^2 \max_{x \in S} \|x\| \prod_{i=1}^{d} \|A^i\|_2^2 \sum_{i=1}^{d} \frac{\|A^i\|_F^2}{\|A^i\|_2^2}}{\gamma^2 m}}\right).$$

solution

$\left|L(f) - \hat{L}(f)\right|$

via compression

$\left|\hat{L}(g) - \hat{L}(f)\right|$ **and** $\left|L(g) - \hat{L}(g)\right|$

Model Compression

Compression for MLP

Analyze MLP:

- **MLP consists of a matrix and ReLU function in each layer**
- **Only the matrix needs compression**

Recall Some Knowledge in Linear Algebra

$$
\begin{aligned}
(a, b, c) &= a(1,0,0) + b(0,1,0) + c(0,0,1) \\
&= a'(1,0,0) + b'(1,1,0) + c'(1,1,1)
\end{aligned}
$$

Where we can call *a*, *b*, *c*, *a'*, *b'*, *c'* the coordinates, and (1,0,0) **etc. the axis directions (or basis vectors); note that axis directions are not unique**

Model Compression

Compression for MLP

Similarly we have:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} = a\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + b\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} + c\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} + d\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$
$$= a'\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + b'\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} + c'\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} + d'\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

Where we can call a, b, c, etc. the coordinates, and $\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$ etc. the axis directions (or basis matrices) ; note that axis directions are not unique

Now what will happen if we have less than $N \times N$ directions?

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \xrightarrow{\text{approximately equal}} \tilde{a}\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} + \tilde{b}\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} + \tilde{c}\begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}$$

We can call that matrix approximation

Model Compression

Compression for MLP

Algorithm 1 Matrix-Project (A, ε, η)

Require: Layer matrix $A \in \mathbb{R}^{h_1 \times h_2}$, error parameter ε, η.
Ensure: Returns $\hat{A}$ s.t. $\forall$ fixed vectors u, v,

$$\Pr[\|u^\top \hat{A} v - u^\top A v\| \geq \varepsilon \|A\|_F \|u\| \|v\|] \leq \eta.$$

Sample $k = \log(1/\eta)/\varepsilon^2$ random matrices $\boxed{M_1, \dots, M_k}$ with entries i.i.d. ± 1 ("helper string")
for $k' = 1$ to k **do**
  Let $Z_{k'} = \boxed{\langle A, M_{k'} \rangle} M_{k'}$.
end for
Let $\hat{A} = \frac{1}{k} \sum_{k'=1}^{k} Z_{k'}$

Basis Matrices

Coordinates

Remarks:

- **The coordinates are the parameters for the compressed model**
- **Number of parameters: $h_1 \times h_2 \rightarrow k$**

Model Compression

Generalization Bound for MLP

Step 1:

- **Prove the approximation is close**

$$\left\|u\left(\hat{A}-A\right)v\right\| \leq \varepsilon\|u\|\|A\|_F\|v\|$$

Step 2:

- **Bound the number of parameters**

$$\frac{72c^2d^2\log(mdh/\delta)}{\varepsilon^2}\cdot\sum_{i=1}^{d}\frac{1}{\mu_i^2\mu_{i\rightarrow}^2}$$

- **Hence a tight generalization bound according to covering number method**

Step 3:

- **(Theorem 4.1) Final generalization bound:**

$$P\left(L_0(f_{\hat{A}}) \leq \hat{L}_\gamma(f_A) + \tilde{O}\left(\sqrt{\frac{c^2d^2}{\gamma^2m}\max_{x\in S}\|f_A(x)\|_2^2\sum_{i=1}^{d}\frac{1}{\mu_i^2\mu_{i\rightarrow}^2}}\right)\right) = 1-\delta$$

Model Compression

Generalization Bound for MLP

Step 3:

- **(Theorem 4.1) Final generalization bound:**

$$P\left(L_0(f_{\hat{A}}) \leq \hat{L}_\gamma(f_A) + \tilde{O}\left(\sqrt{\frac{c^2 d^2}{\gamma^2 m} \max_{x\in S} \|f_A(x)\|_2^2 \sum_{i=1}^{d} \frac{1}{\mu_i^2 \mu_{i\to}^2}}\right)\right) = 1 - \delta$$

- **Comparing with previous bounds:**
- $\prod_{i=1}^{d} \left\|A^i\right\|_2^2$ **is avoided, hence a tight bound**

Theorem 2.2. *((Neyshabur et al., 2017a)) For any deep net with layers $A^1, A^2, \ldots A^d$ and output margin γ on a training set S, the generalization error can be bounded by*

$$\tilde{O}\left(\sqrt{\frac{hd^2 \max_{x\in S} \|x\| \prod_{i=1}^{d} \|A^i\|_2^2 \sum_{i=1}^{d} \frac{\|A^i\|_F^2}{\|A^i\|_2^2}}{\gamma^2 m}}\right).$$

Model Compression

Compression for CNN

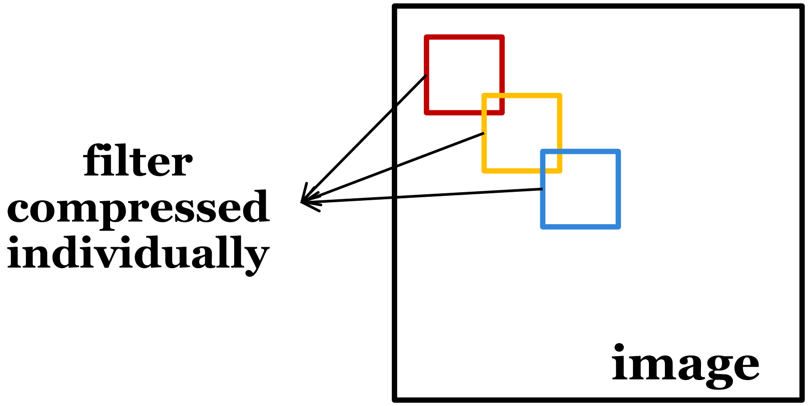
Challenges:

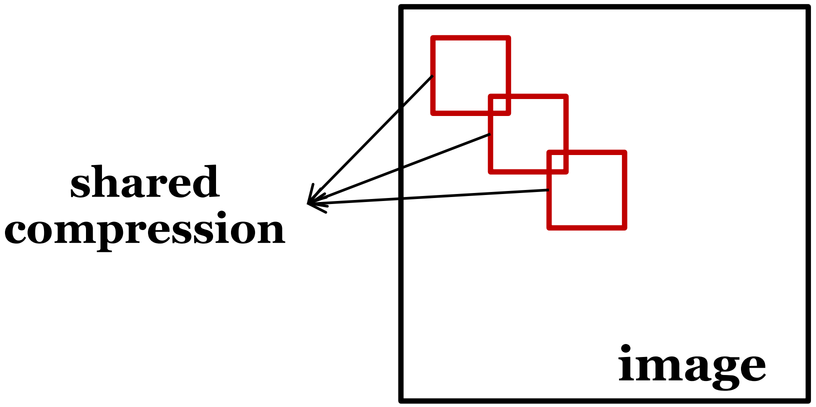
- Filters are shared at different positions on an image
- Cannot naively reuse the algorithm for MLP

Two Trivial Solutions:

- ① Individual compression
- Effect: too many parameters

- ② Shared compression
- Effect: perturbation not independent





Model Compression

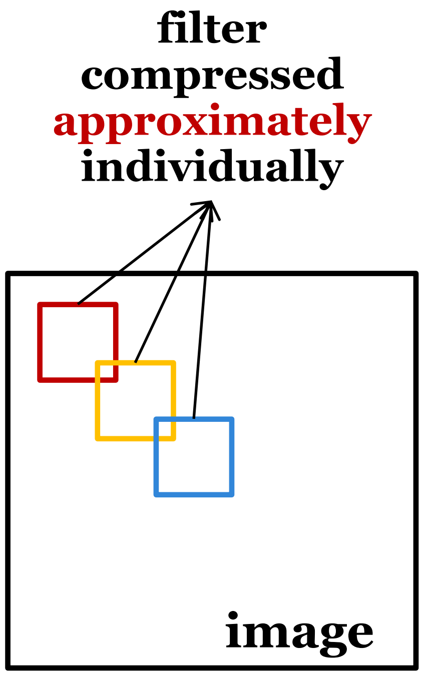
Compression for CNN

Proposed Solution:

- **Generate k × p "basis-basis" matrices randomly, denoted** $\{\breve{M}_{k',p'}\}_{k'\in[1,k],p'\in[1,p]}$
- **At each position, generate basis matrices** $\{M_{k'}\}_{k'\in[1,k]}$ **randomly**
- **Use** $\{M_{k'}\}_{k'\in[1,k]}$ **to express the filter** A
- **Express each** $M_{k'}$ **with** $\{\breve{M}_{k',p'}\}_{p'\in[1,p]}$

Effect:

- **The compression at each position are approximately independent**
- **The number of parameters are limited to k × p, instead of k × n_1 × n_2**



Model Compression

Generalization Bound for CNN

Similar to MLP, we can get:

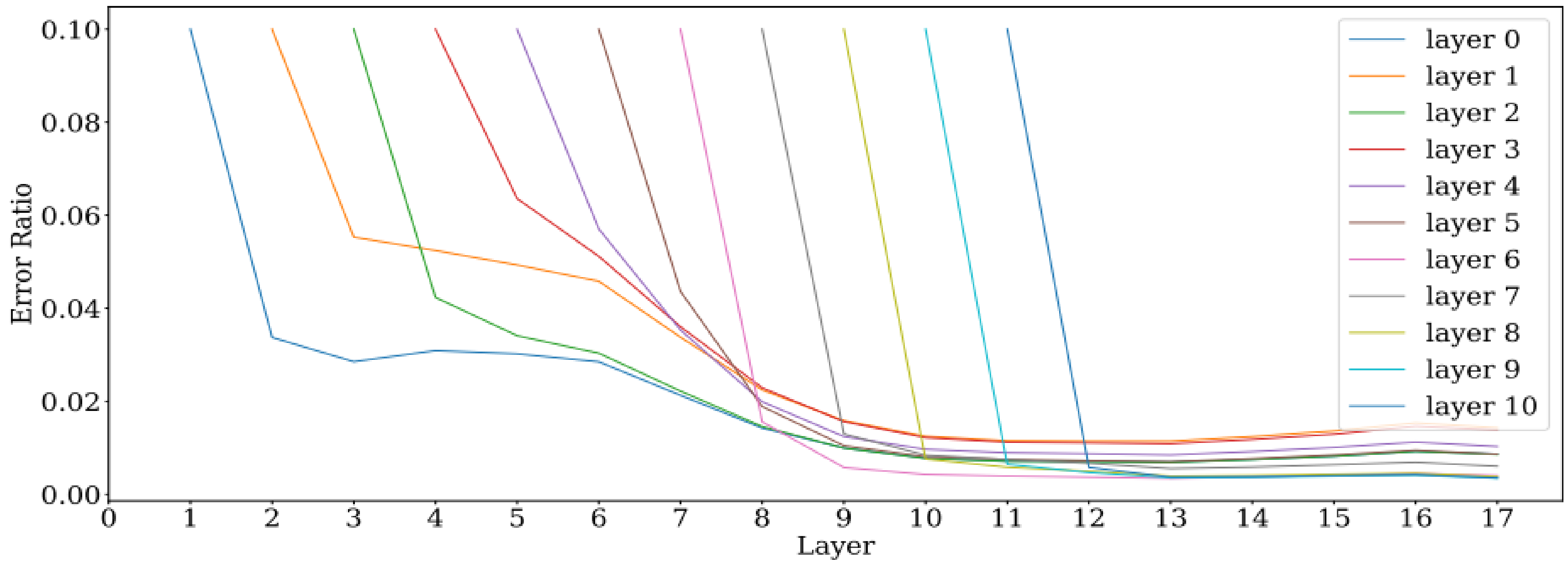
$$P\left(L_0(f_{\hat{A}}) \leq \hat{L}_\gamma(f_A) + \tilde{O}\left(\sqrt{\frac{c^2 d^2}{\gamma^2 m}\max_{x\in S}\|f_A(x)\|_2^2 \sum_{i=1}^{d} \frac{\beta^2 [\kappa_i/s_i]^2}{\mu_i^2 \mu_{i\rightarrow}^2}}\right)\right) = 1-\delta$$

Remarks:

- **The bound is similar to MLP's, except for some extra auxiliary coefficients**
- **Again the product $\prod_{i=1}^{d}\left\|A^i\right\|_2^2$ is avoided, hence a tight bound**

Model Compression

◻ Compression and Noise Stability



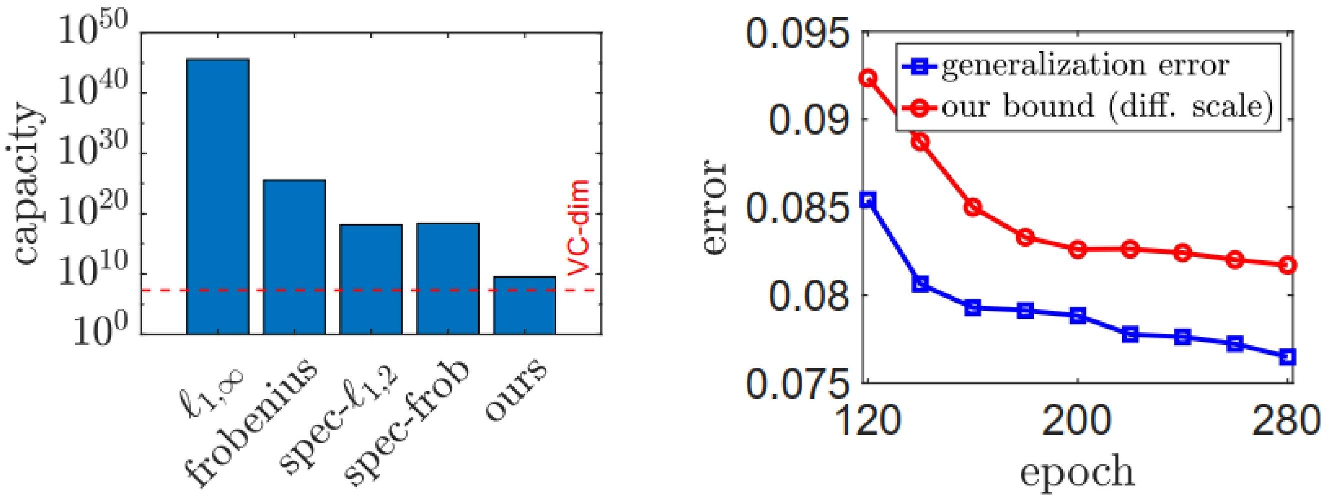
Remarks:

- **Note the attenuating effect of injected noise**
- **Hence the validity of compression (which can be regarded as injected noise)**

Model Compression

☐ Empirical Evaluation

Similar to MLP, we can get:



Remarks:

- **(Left) The proposed bound is indeed tight compared with previous ones**
- **(Right) The proposed bound can improve along with the increasing epoch**

Model Compression

Section Summary

On Compression Methods:

- Compress the model by projecting the matrices into low dimensional space
- Proposed algorithms for MLP and CNN

On Generalization Bounds:

- Derived bounds for MLP and CNN
- Empirically proofed the tightness of the bounds

Outline

- Introduction
- Data Compression
- Model Compression
- Summary

Summary

Compression for Generalization Analysis

Data Compression

- Used a subset of the data set to train and evaluate the model well
- Derived some bounds
- Defined compression schemes and gave some examples

Model Compression

- Proposed compression algorithms and bounds for MLP and CNN
- Analyzed why the bounds are tight
- Empirically evaluated the bounds