Stronger Generalization Bounds for Neural Networks via Compression

Ram Ganesh V and Kishore GV

IIT Madras

Fall 2019

Generalization

- Ability to perform well on unseen data
- We assume both the train data and test data come from the same distribution

Importance of Generalization bounds

- The concepts of generalization error and overfitting is closely related. The more overfitting occurs, the larger the generalization error
- Inspires us to find new algorithms and better regularization techniques

Rademacher Complexity

- Let G be a family of functions from a set $\mathcal{Z}$ to $\mathbb{R}$. Let σ_1, $\ldots$, σ_m be Rademacher variables : $\mathbb{P}(\sigma_i = 1) = \mathbb{P}(\sigma_i = -1) = 1/2$. The empirical Rademacher complexity of G w.r.t. to a sample $S = \{z_i\}_{i=1}^m$ is

$$\mathcal{R}S(G) = \mathbb{E}\sigma \left[\sup g \in G \frac{1}{m} \sum i = 1^m \sigma_i g(z_i) \right]$$

- Gives you an estimate of how much G correlates with noise in S

VC dimension

- The VC-dimension of a hypothesis class H, denoted VCdim(H), is the maximal size of a set C $\subset \chi$that can be shattered by H. If H can shatter sets of arbitrarily large size we say that H has infnite VC-dimension

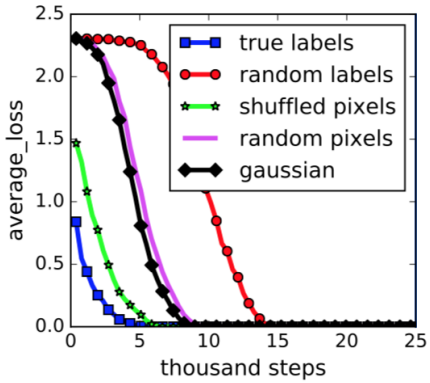
Uniform Complexity

- Uniform stability of algorithm A measures how sensitive the algorithm is to the replacement of a single example
- Solely a property of the algorithm and does not take the distribution of data into account

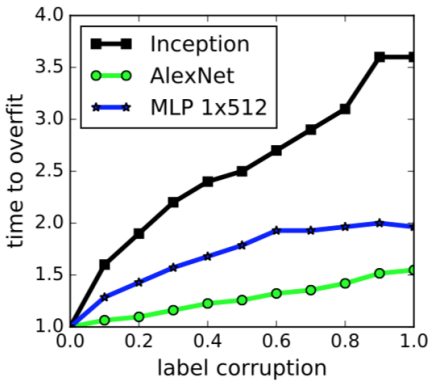
Experiments on Neural networks

- True labels : Original dataset without modification
- Partially corrected labels : independently with probability p, the label of each image is corrupted as a uniform random class
- Random labels : All labels replaced with random labels
- Shuffled pixels : a random permutation of the pixels is chosen and then the same permutation is applied to all the images in both training and test set.
- Random pixels : a different random permutation is applied to each image independently
- Gaussian : A Gaussian distribution (with matching mean and variance to the original image dataset) is used to generate random pixels for each image.

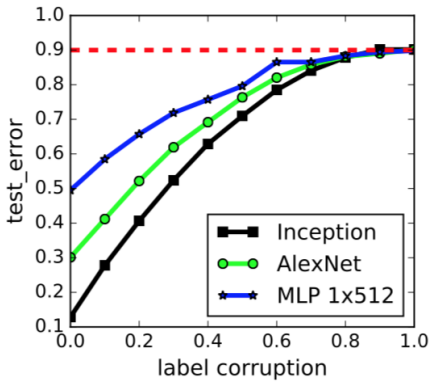
Results



(a) learning curves



(b) convergence slowdown



(c) generalization error growth

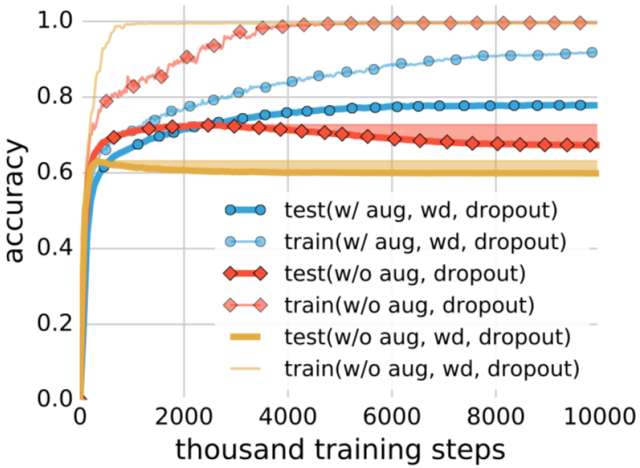
Experiments on Neural networks

- We observe that the effective capacity of neural networks is sufficient for memorizing the entire dataset
- Once the fitting starts, it converges quickly and finally even overfits the training set perfectly
- Random pixels and Gaussian start converging before random labels, partly because input is more separeted from each other than natural images that belong to the same category

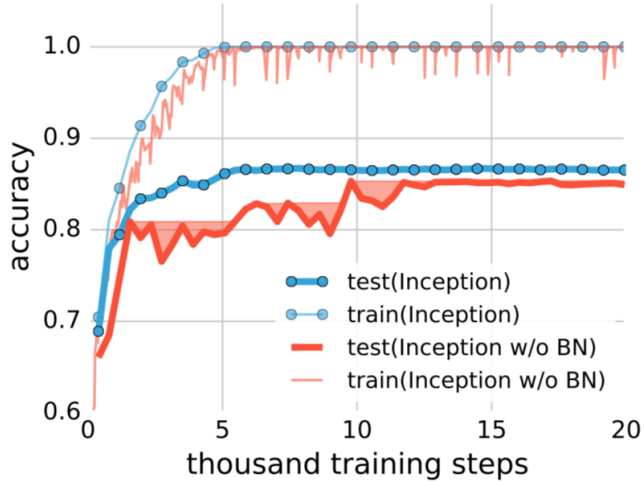
Failures of Existing Generalization Measures

- Since our randomization tests suggest that many neural networks fit the training set with random labels perfectly, we expect that $\Re_n(H) \approx 1$ for the corresponding model class H, which is just a trivial upper bound
- Although there exists a bound on the fat-shattering dimension in terms of l_1 norms, that bound is not relevant for Relu networks.
- Uniform stability is rather pessimistic and thus, we need a weaker notion of stability

More shortcomings of Statistical Learning Theory based measures



(a) Inception on ImageNet



(b) Inception on CIFAR10

More shortcomings of Statistical Learning Theory based measures

- We can clearly see that even with all the regualizers turned off, the model performs very well
- Data augmentation had some significant impact
- Inception achieves good accuracy even without regularization
- The above results seem to suggest that generalization error more or less depends on the data and the model architecture

- Hence, there is a need to develop better ways to bound generalization error of neural networks

Compressing Networks : Idea

- Let the training data contain m samples and let f be a complex classifier with very low empirical bias
- Let g be a much simpler classifier which incurs the same loss on the training data as S. Note that g will generalize much better will incur low classification error on the full distribution
- We try to bound the generalization error of f using the classifier g

Compressing Networks : Methods

- Compression via simple low rank approximation of f
- Compression based on random projections, coupled with noise stability of the network

Compressibility

- Definition 1 ((γ, S)-compressible). Let f be a classifier and $G_{\mathcal{A}} = \{g_A | A \in \mathcal{A}\}$ be a class of classifiers. We say f is (γ, S)-compressible via $G_{\mathcal{A}}$ if there exists $A \in \mathcal{A}$ such that for any $x \in S$, we have for all y

$$|f(x)[y] - g_A(x)[y]| \leq \gamma$$

Bounding Loss

- Suppose $G_{\mathcal{A}}$, $s = \{g_{A,s} | A \in \mathcal{A}\}$ where$A$ is a set of q parameters each of which can have at most r discrete values and s is a helper string. Let S be a training set with m samples. If the trained classfier f is (γ, S)-compressible via $G_{\mathcal{A},s}$ with helper string s, then there exists $A \in \mathcal{A}$ with high probability over the training set,

$$L_0\left(g_A\right) \leq \hat{L}_\gamma(f) + O\left(\sqrt{\frac{q \log r}{m}}\right)$$

An Existing Bound

- (Neyshabur et al.) For a 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

$$\tilde{O}\left(\sqrt{\frac{hd^2 \max x \in S\|x\| \prod i = 1^d \|A^i\| \, 2^2 \sum i = 1^d \frac{\|A^i\|P^2}{\|A^i\|2^2}}{\gamma^2 m}}\right)$$

- We use the following lemma to compress the matrix
- For any matrix $A \in \mathbb{R}^{m \times n}$, let $\hat{A}$ be truncated version of A where singular values that are smaller than $\delta\|A\|^2$ are removed. Then $\|\hat{A} - A\|_2 \leq \delta\|A\|_2$ and$\hat{A}$ has rank at most $|A\|F^2/\left(\delta^2\|A\|_2^2\right)$

Description of the Bound

- The second part of this expression is called stable rank of the layers.Since the square of frobenius norm of a matrix is equal to the sum of the squares of its singular values, and the spectral norm is the square of the largest singular value. Hence, the sum of stable layers os a natural measure of their parameter count
- The first part $\Pi_{i=1}^{d}||A^i||_2^2$ is related to the Lipshitz constant of the network.
- Lipshitz constant is the maximum norm of the vector, it can produce given the input is a unit vector

Method 2

- We introduce some noise stability properties that imply better compression
- We then randomly project linear projections onto lower dimensional subspace

Noise Sensitivity

- If M is a mapping from real-valued vectors to real-valued vectors, and $\mathcal{N}$ is some noise distribution then noise sensitivity of M at x with respect to $\mathcal{N}$, is

$$\psi_{\mathcal{N}}(M, x) = \mathbb{E}_{\eta \in \mathcal{N}} \left[\frac{\|M(x + \eta \|x\|) - M(x)\|^2}{\|M(x)\|^2} \right]$$

- Low sensitivity implies that matrix has very high singular values which can carry the signal X where noise attenuates

Noise Sensitivity

- The noise sensitivity of a matrix M at any vector $x \neq 0$ with respect to Gaussian Distribution $N(0, I)$ is exactly $\frac{||M||_F^2 ||x||^2}{||Mx||^2}$
- The proof is trivial and can be easily derived using the fact $E[\eta\eta^T] = 0$
- The above proposition suggests that if a vector x is aligned to a matrix M, the matrix becomes less sensitive to noise at M

Layer Cushion

- The layer cushion of layer is similarly defined to be the largest number μ_i such that for any $x \in S, \mu_i \left\|A^i\right\|_F \left\|\phi\left(x^{i-1}\right)\right\| \leq \left\|A^i \phi\left(x^{i-1}\right)\right\|$.
- Basically, we're getting a ratio of the norm in data space to the actual norm in the entire space
- Layer cushion can be thought of as a reciprocal to noise sensitivity ($\frac{1}{\mu_i^2}$ in the case of standard Gaussian noise).

Interlayer cushion

- For any two layers $i \leq j$, we define the interlayer cushion $\mu_{i,j}$ as the largest number such that for any $x \in S$:

$$\mu_{i,j} \left\| x^{i^{i,j}} \right\| F \left\| x^i \right\| \leq \left\| J^{i,j}_{x^i} x^i \right\|$$

- Its a more generalized version of the layer cushion term we have defined above

Activation Contraction

- The activation contraction c is defined as the smallest number such that for any layer i and any $x \in S$,

$$\left\|\phi\left(x^i\right)\right\| \geq \left\|x^i\right\| / c$$

- It more or less quantifies the intuitive observation that almost half the layers of Relu are active at a time.
- For Relu, it will approximately 0.5

Interlayer Smoothness

- Let η be the noise generated as a result of substituting weights in some of the layers before layer i . We define interlayer smoothness ρ_δ to be the smallest number such that with probability $1-\delta$ over noise η for any two layers $i<j$ any $x \in S$:
- For one layer
-

$$\left\| M^{i,j}\left(x^i+\eta\right) - J^{i,j}_{x^i}\left(x^i+\eta\right) \right\| \leq \frac{\|\eta\| \left\|x^j\right\|}{\rho_\delta \left\|x^i\right\|}$$

- In order to understand the above condition, we can look at a single layer case where $j = i + 1$:

$$\left\| M^{i+1}\left(x^i + \eta\right) - J_{xi}^{i,i+1}\left(x^i + \eta\right) \right\| =$$

$$\left\| A^{i+1}\phi\left(x^i + \eta\right) - A^{i+1}\left(\phi'\left(x^i\right) \odot \left(x^i + \eta\right)\right) \right\|$$

$$= \left\| A^{i+1}\nu \right\| \leq \frac{\|\eta\| \left\| A^{i+1}\phi\left(x^i\right) \right\|}{\rho_\delta \left\| x^i \right\|}$$

where $\odot$ is the entry-wise product operator and $\nu = \left(\phi'\left(x^i + \eta\right) - \phi'\left(x^i\right)\right) \odot \left(x^i + \eta\right)$. since the activation function is ReLU, $\phi'\left(x^i + \eta\right)$ and $\phi'\left(x^i\right)$ disagree whenever the perturbation has the opposite sign and higher absolute value compared to the input and hence $\|\nu\| \leq \|\eta\|$.

Interlayer Smoothness : Intuition

- In practice, the noise is well distributed and only a small portion of activations change from active to inactive and vice versa. Hence, we can expect $||\eta||$ to be much greater than $||\nu||$.
- In general for a single layer, we see that ρ_δ captures the ratio of input/weight alignment to noise/weight alignment

Johnson Lindenstrauss Lemma

- Unlike PCA, JLS does not assume the data lies constrained to a lower dimensional subspace
- Set up N data points $(u^i)_{i=1}^N$, such that $u^i \in R^d$. We would like to find f s.t $f : R^d \rightarrow R^m$ and $m << d$ while preserving the inter data distance
- $1 - \epsilon \leq \frac{||f(u^i) - f(u^j)||2^2}{||u^i i - u^i j||2^2} \leq 1 + \epsilon$
- s.t $\epsilon > 0$ $\forall (i, j) \in [1, 2...N] \times [1, 2..N]$
- We can prove that $\exists$ a linear map to generate a f which satisfies the above demand such that for $m = \Omega(\frac{logN}{\epsilon^2})$

General Idea on Random Projections

- Perturb the weight matrices by random projections on a lower dimensional subspace
- Using the basic idea of Johnson-Lindenstrauss, which preserves the inter data distances, we can show that the output hasnt changed much
- And finally apply some noise stability properties on the result mentioned in the next slide and bound the Rademacher complexity by Dudley entropy integral

Lemma Based on an Existing Bound

- Let f_A be a d-layer network with weights $A = \left\{A^1, \ldots, A^d\right\}$. Then for any input x, weights A and $\hat{A}$, if for any layer i, $\left\|A^i - \hat{A}^i\right\| \leq \frac{1}{d}\left\|A^i\right\|$, then we have :

$$\left\|f_A(x) - f_{\hat{A}}(x)\right\| \leq e\|x\| \left(\prod_{i=1}^{d} \left\|A^i\right\| 2\right) \sum i = 1^d \frac{\left\|A^i - \hat{A}^i\right\| 2}{\left\|A^i\right\| 2}$$

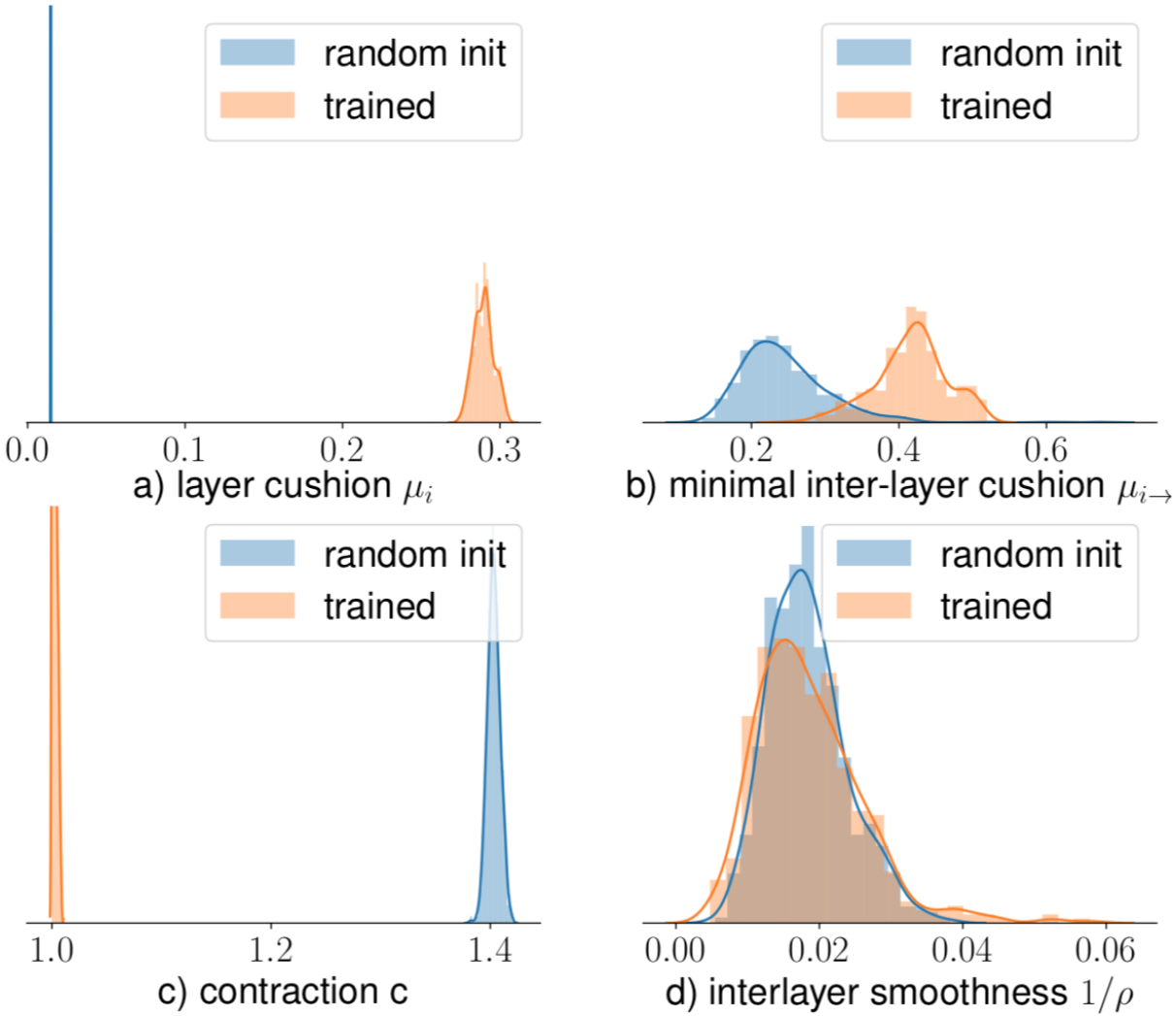
Getting the final equation

- $$\beta^d = \prod_{i=1}^{d} \|A^i\| \leq \frac{c\|x^1\|}{\|x\|\mu_1} \prod_{i=2}^{d} \|A^i\| \leq \frac{c^2\|x^2\|}{\|x\|\mu_1\mu_2} \prod_{i=2}^{d} \|A^i\|$$

$$\leq \frac{c^d \|f_A(x)\|}{\|x\| \prod_{i=1}^{d} \mu_I}$$

- We compute the new weights by approximating all the parameters by ν

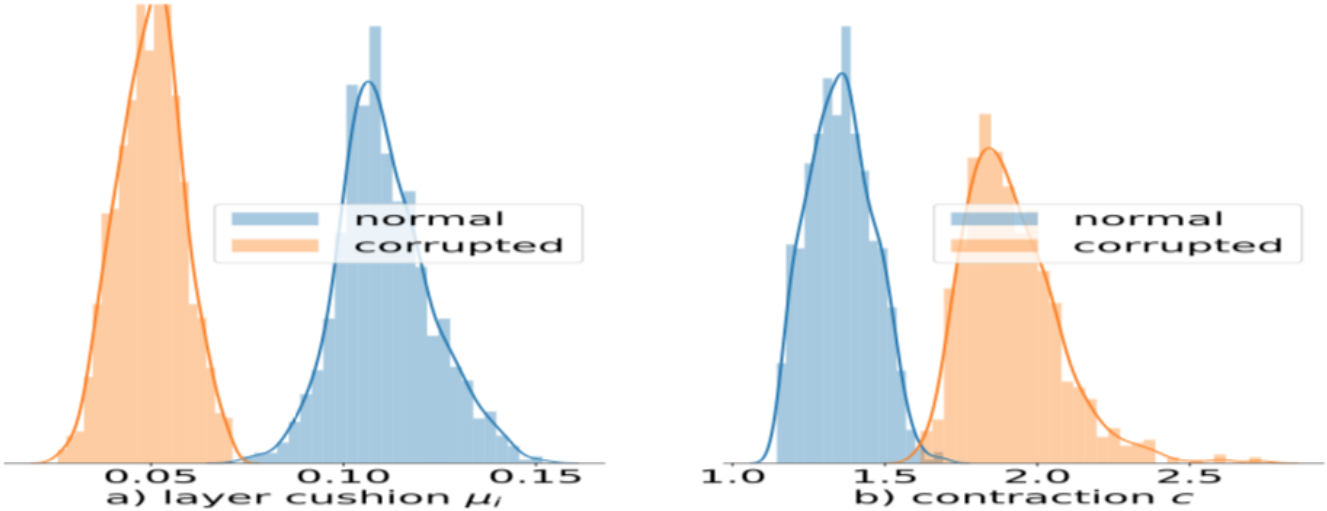
Inferences


random init
trained
0.0
0.1
0.2
0.3
a) layer cushion μ_i
random init
trained
0.2
0.4
0.6
b) minimal inter-layer cushion $\mu_{i\to}$
random init
trained
1.0
1.2
1.4
c) contraction c
random init
trained
0.00
0.02
0.04
0.06
d) interlayer smoothness $1/\rho$

Empirical Investigation of Noise

- We observe that layer cushion is the driver feature which makes neural networks generalize better
- We also observe that c decreases after training, which makes intuitive sense as a trained network will have better generalization power than an untrained one
- There is some overlap in the inter layer cushion graph. This is because, even in untrained networks, aggregation of layers can cancel out some effects of noise.
- Inter layer smoothness doesn't seem to have much of an impact towards helping a neural network generalize. This is because as we have seen earlier, the bounds are already quite good.

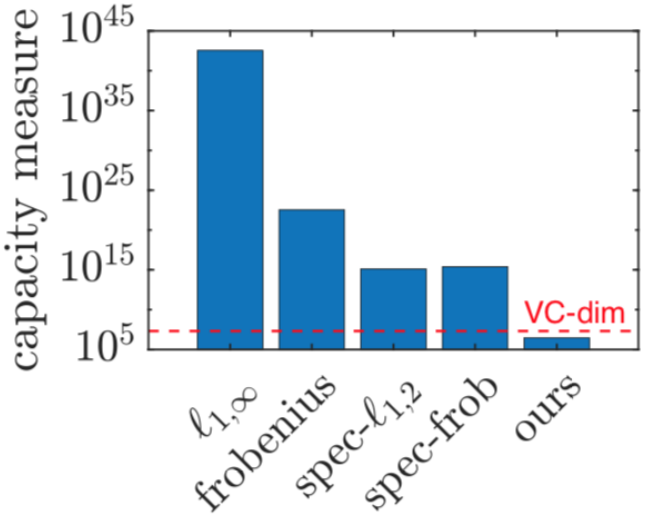
Normal data and corrupted data



Observe that the trained network has a larger cushion and a lower activation contraction.

Comparison To Other Generalization Bounds

layer	$\frac{c_i^2\beta_i^2\lceil\kappa_i/s_i\rceil^2}{\mu_i^2\mu_{i\to}^2}$	actual # param	compression (%)
1	1644.87	1728	95.18
4	644654.14	147456	437.18
6	3457882.42	589824	586.25
9	36920.60	1179648	3.129
12	22735.09	2359296	0.963
15	26583.81	2359296	1.126
18	5052.15	262144	1.927



The new bound is order of magnitudes better

We also observe that the compression in the earlier layers are very high, but are very low in the later layers.

Conclusion and Future Work

- Explored two different compression frameworks invorder to get better generalization bounds for neural networks
- Tried to gain an intuitive understanding on the various noise stability measures used
- We got tighter bounds on some existing results
- We can also extend this to CNNs
- Latest work uses something called Stable Rank Normalization, which minimizes the stable rank and gets much better generalization bounds