A Survey on the Law of Robustness

Mathematics of Deep Learning, Spring 2022
Namas Mankad - nkm9348
New York University

zoom presentation link

Deep neural networks show a lack of robustness to small perturbations. Deep neural networks are trained with many more parameters than the number of training data points and the empirical studies show that increasing the *size* of the model helps in better memorization and generalization. Which is in contrast to the classical theory which would suggest that data interpolation with a parameterized model class is possible as long as the number of parameters is larger than the number of equations to be satisfied. Alternatively, one only requires n equations to solve for n unknowns. The law of robustness proves that for very general class data distributions and model classes, overparameterization is necessary if one wants to fit data in a *smooth or robust* way, as measured by the Lipschitz constant. It shows inherent trade-off between the size of the neural network and its robustness. More specifically, we need the number of parameters in the order of n times d where n is the number of data points and d is the dimension. The law states true as long as the parametrized function class is *smooth*, weights are polynomially bounded and the covariate distribution supports the isoperimetry property.

1. Introduction

Robustness is a crucial attribute for a neural network that defines how well the given network can perform under tiny perturbations. Deep neural networks have been shown to lack robustness to small input changes through adversarial input generation. Despite their widespread application, they fall short on their use in safety-critical systems. There exists a rich amount of literature where a network trained to high accuracy on a given data set is "swindled" by adversarial attacks which are inputs generated that differ only slightly from the inputs in the training set. Thus, adversarial analysis of a neural network has become a very important part of the process of formal verification of the network.

We know from classical regimes that one can solve n equations from n unknowns. But Deep learning methods involve highly parameterized models where even though we have only n data points to learn from, the number of parameters far surpass n. Moreover, the models seem to work better as the *size* (parameters) increases. The law of robustness is an explanation of this phenomenon/observation for a very broad class of distributions and models classes. Informally, it shows that finding a *smooth* function to fit d-dimensional, n data points one needs at least nd parameters. Which means finding a smooth function that perfectly fits the data below noise level in a class of function is much more difficult. We can also say that we need d times more parameters(overparameterization) for *smooth* interpolation. Formally, for any function class smoothly parametrized by p parameters, for any d dimensional dataset satisfying mild regularity conditions, any function that fits the data below the noise level must have its Lipschitz constant to be at least $\sqrt{\frac{nd}{p}}$.

1.1 Lipschitz as a notion of smoothness

Definition 1: If *f*: $R^d \rightarrow R$ is a function which is L-lipschitz i.e $Lip(f) \leq L$, L defines how robust it actually is

$$| f(x) - f(x') | \leq L \,\|x - x'\|$$

For the models that do not perform well in the case of adversarial attacks or slight perturbations, we see that they have a large Lipschitz constant (section 2, 3). But why do we care about a function being Lipschitz in the first place? From definition 1 we see that for a *small* change in the input the output changes proportionally by a similar *small* amount. Intuitively, if one could train such a neural network with a small lipschitz constant then one would train a much smoother function that is not over sensitive to small perturbations. We take Lipschitz as a measure or norm for robustness.

1.2 Benign overfitting and empirical analysis

We look at several models trained on MNIST and CIFAR-10, and observe the empirical results presented in [8], [11]. Experiment from [11] that trains a ResNet-18 model on the CIFAR-10 dataset with 15% label noise and observes the plot between the test error and the size of the model as shown in fig. 1. We see a very general phenomenon explained by the classical learning theory that as the size of the model increases the test error decreases and the model generalizes but after reaching a minimum the model starts overfitting as the size increases. But as we increase the size further and exit the early stopping regime, we see a "double descent" phenomenon where the test error starts to decrease again. The test error decreases even more than the initial minima achieved. We see similar results for experiments conducted in [12][13][14].

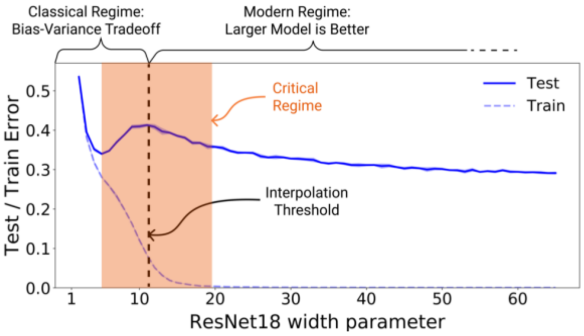

Fig 1. "Double descent": Size of the model vs test error on cifar-10 with 15% label noise

Looking at the empirical results presented in [8] we see that for the model trained on MNIST with PGD-made adversarial examples, the model starts to generalize better after a certain threshold of number of parameters is reached. At roughly $p \simeq 10^6$ we see a "phase change" where the model starts learning better as shown in fig 2. The law of robustness finds a framework to explain such behavior of high dimensional data on large models.

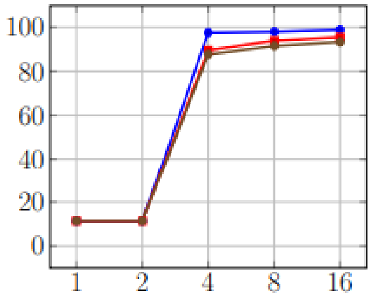

Fig 2. Size of the model vs test error on mnist with PGD adversarial attacks

1.3 Memorization, model class and size

The size of a function class F is measured by the number of parameters p needed to find/pick an individual function f in this class. Each function in our class is defined by a parameter vector of p real values and that vector is polynomially bounded so that the function does not behave abruptly. Formally, We measure "size" as: $f_v \in F$, where $v \in R^p$, $|v| \leq poly(d)$. We study two-layer neural networks with inputs in R^d, k neurons, and Lipschitz non-linearity $\psi : R \rightarrow R$. The function class is as follows:

$$x \mapsto \sum_{l=1}^{k} a_l \psi\left(w_l . x + b_l \right), \; a_l, b_l \in R, \; w_l \in R^d \tag{1}$$

with a_l , $b_l \in R$ and $w_l \in R^d$ for any $l \in [k]$. $F_k(\psi)$ is the notation for the functions of the form (1). Now, the memorization task at hand is; that for a given data set $(x_i , y_i) \in R^d \times R$ where $i \in \{1,...,n\}$, one can find $f \in F_k(\psi)$ such that

$$f(x_i) = y_i \;\; \forall \; i \in [n] \tag{2}$$

or,

$$\sum \left(f\left(x_i \right) - y_i \right)^2 \leq \frac{1}{c} \sum_i z_i^2, \; c > 1 \tag{3}$$

where z_i is the noise if $y_i = g(x_i) + z_i \; \forall \; i \in [n]$. Alternatively, we can say that memorization in a robust way is to optimize the training error below the label noise level. So the model learns at least some of the noise. The reason behind taking this class of function is because it is a very general and universal class of functions. Generally with a large enough value of k we can satisfy any memorization task of any given dataset given the non linearity ψ is not polynomial[7][9]. If ψ is polynomial of, say, degree p, and if the data points are considered i.i.d on the sphere, one could only memorize d^p points. In a variety of scenarios one is furthermore interested in fitting the data smoothly. For example, in machine learning, the data fitting model f is used to make predictions at unseen points $x \notin \{x_1, . . . , x_n\}$. It is reasonable to ask for these predictions to be stable, that is a small perturbation of x should result in a small perturbation of $f(x)$.

1.4 Isoperimetry

Isoperimetry and concentration of measure are the key properties of a high dimensional space. It asserts in many cases that lipschitz functions on high dimensional space concentrate tightly around their mean. Assume that the distribution μ of the covariates x_i satisfy such an inequality in the following sense:

Definition 2: A probability measure μ on R^d satisfies satisfies *c-isoperimetry* if for any bounded L-lipschitz function $f\colon R^d \rightarrow R$, and any $t \geq 0$,

$$P^{\mu}\left[\left|f(x) - E^{\mu}[f]\right| \geq t\right] \leq 2e^{\frac{-dt^2}{2cL^2}}$$

If we look at the values of any lipschitz function f on an input drawn from the distribution μ then the value f has some sub-gaussian tails. The factor of d in the exponent of definition 2 shows that the tail decay will be more and more extreme as we go in the higher dimension. In general, if a scalar random variable X satisfies $P[|X| \geq t] \leq 2exp(-t\ 2/C)$ then we say X is C-subgaussian. Distributions satisfying $O(1)$-isoperimetry include high dimensional Gaussians $\mu = N\left(0, \frac{I_d}{d}\right)$ and uniform distributions on spheres and hypercubes. Isoperimetry also holds for manifolds with Ricci curvature. The section 6 discusses the most generalized version of the law of robustness which poses to be realistic from the perspective that under manifold-hypothesis high dimensional data tends to lie on a lower dimensional manifold. Theorem 5 in section 6 which presents the universal law of robustness, also applies to distributions formed as a mixture of other isoperimetry satisfying distributions.

2. Corollary for the conjecture in section 4

How large does k need to be able to memorize arbitrary labels? We look at a construction presented in [4] that shows,

Theorem 1: If $\psi(x)=\mathbb{1}\{x \geq 0\}$ is a step function, for binary labels and data in general position, $k \geq 2n / d$ suffice to memorize. [4]

Alternatively if p is the number of parameters, required $p \simeq n$. The construction says that one can take a thin region around an affine subspace to fit d data points with the same labels. The region Indicator $\mathbb{1}$ of a slab around $w.x = t$, is given by just the difference of 2 neurons:

$$\mathbb{1}\{w.x \geq t - \varepsilon\} - \mathbb{1}\{w.x \geq t + \varepsilon\} \qquad (4)$$

This gives us value 1 in that region but 0 everywhere else. Since the data is in general position, we can make the region thin enough so that it does not contain any other points. So we need n/d such regions. Since we need 2 neurons to describe each region, we get $k \geq 2n/d$. Similarly, for $\psi(x) = ReLU(x)$ we get $k \geq 4n/d$. But one can see that this construction overfits since it just gives value on those d data points and does not generalize. Although this construction can memorize in

$k \simeq n/d$, it is very non-ideal since its Lipschitz constant is very high. We derive its lipschitz constant as follows:

We consider a well dispersed data x_i where $i \in \{1,...,n\}$ and $n=poly(d)$, is i.i.d on the sphere S^{d-1}, i.e uniformly distributed vectors with norm 1, we take d points and make them go through a subspace. The inner product t in $w.x = t$, is going to be at the distance $d_1 = 1/\sqrt{d}$, since the mass is concentrated in that width d_1. So a particular neuron is at distance d_1 from the origin. But in high dimension, all of the x_i are going to be in width d_1 so it means that the width of my baum construction has to be d_1. And since the width of the thin region is $d_1 = 1/\sqrt{d}$, one can go from value 0 to 1 in that same distance. So the Lipschitz constant will be at least $\sqrt{d}$.

$$Lip \geq \sqrt{d} \tag{5}$$

This is a poor lipschitz constant since it scales with the dimension. One more thing to note is that the data being in general position is important because otherwise all the data points could lie on a line. Then, one cannot hold the change in dimension to have an improvement. Now one could also add *infinitesimally small* noise which will put the data points in general position but then the thn region will have to be even more *thin* in width and hence the lipschitz constant will be even higher.

3. Constant lipschitz construction

We have a construction that can memorize the data in constant lipschitz. For the distributions μ we have in mind, for instance uniform on the unit sphere, there exists with high probability some O(1)-Lipschitz function $f : R^d \rightarrow R$ satisfying $f(x_i) = y_i$ for all i. Indeed, with probability $1- e^{-\Omega(d)}$ we have $||xi - xj|| \geq 1$ for all $1 \leq i \neq j \leq n$ so long as $n \leq poly(d)$. In this case we may apply the Kirszbraun extension theorem to find a suitable f regardless of the labels y_i . More explicitly we may fix a smooth bump function $g : R^+ \rightarrow R$ with $g(0) = 1$ and $g(x) = 0$ for $x \geq 1$, and then interpolate using the sum of radial basis functions or bump functions

$$f(x) = \sum_{i=1}^{n} g(||x - x_i||)y_i. \tag{6}$$

.

In fact this construction requires only $p = n(d + 1)$ parameters to specify the values $(x_i , y_i)_{i\in[n]}$ and thus determine the function f. Hence $p = n(d + 1)$ parameters suffice for robust interpolation, i.e. Theorem 3 in section 6 is essentially best possible for $L = O(1)$. A similar construction shows the same conclusion for any $p \in [\Omega'(n), nd]$, essentially tracing the entire tradeoff curve. This is because one can first project onto a fixed subspace of dimension $\tilde{d} = \frac{p}{n}$, and the projected inputs x_i now have pairwise distances at least $\Omega\left(\sqrt{\frac{\tilde{d}}{d}}\right)$ with high probability. The analogous construction on the projected points now requires only $p = \tilde{d}n$ parameters and has Lipschitz constant,

$$O\left(\sqrt{\frac{d}{\tilde{d}}}\right) = O\left(\sqrt{\frac{nd}{p}}\right). \tag{7}$$

We can interpolate/memorize polynomially many points on the sphere with constant lipschitz. For *poly(d)* data points, those points will be very isolated on the sphere. In high dimensions, if we visualize a smaller sphere/ball on a particular data point then there will be no other data points in that sphere. So to interpolate smoothly we can apply a bump function/radial basis function at each of the data points. Bump function has continuous derivatives of all orders, and so we can go from 0 to 1 in a small distance. Since these functions are isolated as well we can add them of given height and the label that one wants to see and it will have a constant lipschitz. So we can have the number of neurons in the order of the number of datapoints. This bump function can be approximated with constant neurons and we need a bump function for each data point so we need order of n neurons. Hence, we get $k \simeq n$ or alternatively, $P = nd.$

4. Trade off between size and smoothness

One has two choices, either create a Baum construction([4], section 2) that constructs a small model with $k \simeq n/d$, *i.e* $(P = n)$ but has a much worse lipschitz constant with $Lip \simeq \sqrt{d}$. Alternatively, create a much larger model with $k \simeq n$, *i.e* $(P = n*d)$ but has a much better and desired lipschitz constant $Lip \simeq 1.$

Now, we can propose the theorem of robustness as a conjecture and discuss proofs of their sub optimal versions in section 5.

Conjecture 1: For generic data sets, with high probability, any $f \in F_k(\psi)$, *where* p are the number of parameters fitting the data (i.e., satisfying (2) or (3)) must also satisfy:

$$Lip_{S^{d-1}}(f) \geq c\sqrt{\frac{n}{k}} \text{ or alternatively, } Lip(f) \geq \sqrt{\frac{nd}{P}} \tag{8}$$

Note that for generic data, with high probability (for $n = poly(d)$), there exists a smooth interpolation. Namely there exists $g : R^d \rightarrow R$ with $g(x_i) = y_i$, $\forall\ i \in [n]$ *and* $Lip(g) = O(1)$. This follows easily from the fact that with high probability (for large d) one has $||x_i - x_j|| \geq 1,\ \forall\ i \neq j.$ Conjecture 1 puts restrictions on how smoothly one can interpolate data with small neural networks. A striking consequence of the conjecture is that for a two-layers neural network $f \in Fk(\psi)$ to be as robust as this function g (i.e., $Lip(f) = O(1)$) and fit the data, one must have $k = \Omega(n)$, i.e., roughly one neuron per data point. On the other hand with that many neurons it is quite trivial to smoothly interpolate the data. Thus the conjecture makes a strong statement that essentially the trivial smooth interpolation is the best thing one can do. In addition to making the prediction that one neuron per datapoint is necessary for optimal smoothness(section 3), the conjecture also gives a precise prediction on the possible tradeoff between size of the network and its robustness.

5. Proofs of sub optimal versions of the conjecture

Conjecture 1 can be made severely weaker along several directions. For example the quantity of interest $Lip_{Sd-1}(f)$ can be replaced by a quantity that depends on the spectral norm of the weight matrix (essentially ignoring the pattern of activation functions). For that proxy, see Theorem 2, which in particular formally proves that “overparameterization is a law of robustness for generic data sets". Other interesting directions to weaken the conjecture include specializing it to common activation functions, or simply having a smaller lower bound on the Lipschitz constant. In section 5.2 we discuss the conjecture proved in "under complete case" i.e., n is replaced by d in the lower bound. So the conjecture's lower bound is proved for $\sqrt{d/k}$ instead of $\sqrt{n/k}$. It matches the conjecture for $n \approx d$, in the sense that only $k \leq d$ is relevant. For the case of $n \gg d$ i.e., moderately high dimensional case, the proof cannot work. Lastly in section 5.3 we discuss the conjecture proved in the optimal size regime i.e., $k.d \approx n$ for polynomial activation functions first with degree of the activation function p=2 i.e., quadratic and then for a general polynomial activation.

5.1 Statistical approach - spectral norm proxy for the lipschitz constant

We rewrite (1) as

$$f(x) = a^\top \psi(Wx + b)\,, \tag{9}$$

where $a = (a_1, \ldots, a_k) \in R^k$, $b = (b_1, \ldots, b_k) \in R^k$, $W \in R^k{\times}d$ is the matrix whose l^{th} row is w_l , and ψ is extended from $R \to R$ to $R^k \to R^k$ by applying it coordinate-wise.

Theorem 2: Assume that ψ is L-Lipschitz. For $f \in F_k(\psi)$ one has

$$\mathrm{Lip}(f) \leq L \cdot \|a\| \cdot \|W\|_{\mathrm{op}}\,. \tag{10}$$

For a generic data set, if $f(x_i) = y_i$, $\forall\, i \in [n]$ and f has no bias terms (i.e., $b = 0$ in (9)), then with positive probability one has:

$$L \cdot \|a\| \cdot \|W\|_{\mathrm{op}} \geq \sqrt{\frac{n}{k}}\,. \tag{11}$$

5.2 Geometric approach - the under complete case

Theorem 3: Let $n \geq d$. Let $f : R^d \to R$ be a function such that $f(x_i) = y_\mathrm{i}$, $\forall\; i \in [n]$ and moreover $f(x) = g(Px)$ for some differentiable function $g : R^k \to R$ and matrix $P \in R^{k\times d}$. Then, for generic data, with probability at least $1 - exp(C - cd)$ one must have

$$Lip(f) \geq c\sqrt{\frac{d}{k}} \tag{12}$$

This lower bound applies to much more general models, mainly multi index models. Single index model is just high dimensional data but one makes decisions based on the projection in one direction. Since we need to think about multi index models, we have k directions. So, P is a projection on a k dimensional subspace, and the theorem claims that if f memorizes random data on a sphere then it must be the case that (12) holds true. Intuitively, for any fixed k-dimensional subspace, and most directions $\theta \in S^{d-1}$, one has $||P\theta||_2 \simeq \sqrt{k}/\sqrt{d}$. So if there is such a direction θ in the dataset where f goes from -1 to 1, then it means that f also goes from -1 to 1 between $-P\theta$ and $P\theta$, but the length of that segment is only $\sqrt{k}/\sqrt{d}$, so f is at best $\sqrt{d}/\sqrt{k}$ lipschitz. We have a sphere in dimension d and if the sphere is projected in one direction, we know that most of the mass is going to be concentrated in radius *1/√d*. So if the sphere is projected in *k* dimensional subspace, in total we would have shrunk the sphere to radius

$$\sqrt{\frac{1}{\sqrt{d^2}}+\frac{1}{\sqrt{d^2}}+\ldots k\ times} = \sqrt{\frac{k}{d}} \tag{13}$$

Since, one has to *move* on a much smaller sphere, we see that (9) holds true.

5.3 Algebraic approach - polynomial activation

Theorem 4: Assume that we have a tensor *T* of order p such that $\langle T, x^{\otimes p}{}_i\rangle = y_i$, $\forall i \in [n]$. Then, for generic data, with probability at least $1 - C\ exp(-c_p d)$, one must have,

$$\|T\|_{op} \geq c_p\sqrt{\frac{n}{d^{p-1}}} \tag{14}$$

Denoting $\Omega = \Sigma_{i=1}{}^n\, y_i x_i^{\otimes p}$, we have (using $y_i^2 = 1$ for the first equality and $||T||_{op,*} \leq d^{p-1} \cdot ||T||_{op}$ for the last inequality):

$$n = \langle T, \Omega\rangle \leq\ ||\Omega||_{op} \cdot ||T||_{op,*} \leq d^{p-1} \cdot ||\Omega||_{op} \cdot ||T||_{op} \tag{15}$$

Thus we obtain,

$$\|T\|_{op} \geq \frac{n}{d^{p-1}.\|\Omega\|_{op}} \tag{16}$$

and with probability at least $1 - C\ exp(-c_p d)$ one has,

$$\|\Omega\|_{op} \leq C_p\sqrt{\frac{n}{d^{p-1}}} \tag{17}$$

For quadratic activations:

With $\psi(t) = t^2$, one hidden layer neural network can shown as $f(x) = \sum_{l=1}{}^k a_l.\psi(w_l.x)$, which is $f(x) = \sum_{l=1}{}^k a_l.(w_l.x)^2$ which can be written as the hilbert-schmidt inner product between some matrix *Σ* and the outer product xx^T and so we get $f(x) = \langle \Sigma, xx^T\rangle$, where $\Sigma = \sum_{l=1}{}^k a_l.w_l.w_l^T$. We don't think of individual neurons but *Σ* specifies the underlying matrix that characterizes the one hidden layer neural network. Also note that $Lip_{sd-1}(f) = 2||\Sigma||_{op}$. Consider a *small* quantity $\Omega = \sum_{l=1}{}^n y_i x_i x_i^T$ which can be thought of as a random walk over the rank 1 matrices, we can have $\pm x_1x_1^T,\ \pm x_2x_2^T,\ \pm x_3x_3^T$ …. and so on. Intuitively we see a lot of "*cancellations*" in the terms and hence the value of *Ω* is s*mall.* We now prove that the tensor that represents the network i.e the matrix *Σ* during memorization correlates with *Ω*. We also have that $n = \langle \Sigma, \Omega\rangle \leq ||\Sigma||.||\Omega||_*$ for any pair of dual norms. And since *Ω* is *small* it must be the case that *Σ* is big to compensate. In other words, operator norm $||\Sigma||$ is lower bounded by *n* over the dual of the operator norm i.e nuclear norm $||\Omega||_*$. Nuclear norm is the sum of the absolute value of the eigenvalues. It is easy to see that $||\Omega||_{Fr} \lesssim \surd n$ which implies, $||\Sigma||_{Fr} \gtrsim \surd n$. But we already know that $||\Sigma||_{Fr} \leq ||\Sigma||_{op}.\surd k$. So we get, $||\Sigma||_{op} \gtrsim \surd n/\surd k$, Σ is rank k and we prove theorem 4 and the conjecture 1 for quadratic activation functions.

Even better, one can obtain an even bigger lower bound on the lipschitz constant by studying the $||\Omega||_{op}$ as shown,

$$\|\Sigma\|_{op} > \sqrt{\frac{n}{k}} \cdot \sqrt{\frac{d}{k}}$$

For degree p, we cannot say that $||T||_{Fr} \leq ||T||_{op}.\surd k$, but this inequality holds for the maximum rank case where $k = d^{p-1}$, and hence we can say that (14) holds true.

6. Universal law of robustness

For the function class of two-layer neural networks, we have discussed many approaches to prove the law of robustness. The strategies of proving relied on various ways to measure how *large* the set of two-layer neural networks can be. We have seen a statistical approach based on rademacher complexity (section 5.1), geometric approach based on relation to the multi-index models (section 5.2) and algebraic approach in the case of polynomial activations (section 5.3). Now, we discuss a different approach where we focus on an individual function $f \in F$ instead of the function class F. In other words, for a fixed function f, what is the probability that it would give a good approximate fit on the random data?

Theorem 5: . Let F be a class of functions from $R^d \rightarrow R$ and let $(x_i,y_i)_{i=1}{}^n$ be i.i.d. input-output pairs in $R^d \times [-1, 1]$. Fix , $\delta \in (0, 1)$. Assume that:

1. The function class can be written as $F = \{f_w, w \in W\}$ *with* $W \subset R^p$, $diam(W) \leq W$ *and for any* $w_1, w_2 \in W$,
$$\left\| f_{w_1} - f_{w_2} \right\|_{\infty} \leq J \left\| w_1 - w_2 \right\|. \tag{18}$$
2. The distribution μ of the covariates x_i can be written as $\mu = \sum^k_{l=1} \alpha_l \mu_l$, where each μ_l satisfies *c-isoperimetry*, $\alpha_l \geq 0$, $\sum^k_{l=1} \alpha_l = 1$, and k is such that $9^4 k \log(8k/\delta) \leq n\varepsilon^2$.
3. The expected conditional variance of the output is strictly positive, denoted $\sigma^2 := E^\mu[Var[y|x]] > 0$.

Then with probability at least 1−δ over the sampling of the data, one has simultaneously for all $f \in F$:

$$\frac{1}{n} \sum \left(f\left(x_i \right) - y_i \right)^2 \leq \sigma^2 - \varepsilon \Rightarrow Lip(f) \geq \frac{\varepsilon}{2^9 \sqrt{c}} \sqrt{\frac{nd}{p \log\left(60WJ\varepsilon^{-1} \right) + \log\left(\frac{4}{\delta} \right)}} \tag{19}$$

More simply,

$$\frac{1}{n} \sum \left(f\left(x_i \right) - y_i \right)^2 \leq \sigma^2 - \varepsilon \text{ which implies } Lip(f) \geq \Omega\left(\sqrt{\frac{nd\varepsilon^2}{p\log(BJ)}} \right) \tag{20}$$

In simpler terms, assumption 1 in theorem 5 says that F admits a *J-lipschitz* parametrization by *p* real parameters, each of size *B=poly(n,d)* which means that this parametrization is lipschitz. We also want that *J* and *B* enter the bound logarithmically. Assumption 2 states that we do not need to assume that the distribution of the input is i.i.d on the sphere. Assumption 3 assumes label noise. If these assumptions hold true then with high probability we can say that for any function *f*, if the training error is below σ^2 by some gap ε, where σ and ε can be thought of as constants then the lipschitz constant has to larger than $\sqrt{\frac{nd}{p}}$.

Let us consider a case where we require *f* to perfectly fit or interpolate the data and say that y_i are random ±1 labels. Then with high probability, at least ⅓ of the labels are +1 and at least ⅓ of the labels are -1. The key insight is that isoperimetry implies that either the 0-level set of *f* or the 1-level set of *f* must have probability smaller than $\exp\left(-\frac{d}{Lip(f)2}\right)$. If we were to ignore the data $x_1 \ldots x_n$, isoperimetry says that the function *f* has a very *small* amount of the sphere where it is +1 or it has a very small amount of sphere where it is -1. +1 or -1 has to have an exponentially small probability of *exp(-d)* in the dimension *d*. One could make a function that is +1 on one half of the sphere and -1 on the other half of the sphere by making it discontinuous. Or one could make it continuous in the thin slab in between but isoperimetry property of high dimension space says that for such a construction the thin *slab* in between where the value changes from -1 to +1 has to be very thin and thus it will have a large lipschitz constant. Moreover, if we want f to output the unlikely label on *many* points of the input then that takes the probability to a further exponentially smaller value in the order of *exp(-nd)*. Precisely, the probability that *f* fits all the *n* points is at most $\exp\left(-\frac{nd}{Lip(f)2}\right)$ so long as both labels $y_i \in \{-1,+1\}$ actually appear a constant fraction of the time. In particular, using an union bound, for a finite function class *F* of size N with L-Lipschitz functions, the probability that there exists a function $f \in F$ fitting the data is at most,

$$N\, exp\left(-\frac{nd}{L^2}\right) = exp\left(\log(N) - \frac{nd}{L^2}\right) \tag{21}$$

Thus we see that if $L \ll \sqrt{\frac{nd}{log(N)}}$, then the probability of finding a fitting function in *F* is very small. For a smoothly parametrized family with *p* (bounded) parameters, through a standard discretization argument, one can say that *log(N)=O(p).*

7. Generalization

In previous sections we discuss mainly about memorization, now we discuss how the law of robustness can be phrased in a slightly stronger way, as a generalization bound for classes of lipschitz functions based on the data-dependent rademacher complexity. We also recall that small Rademacher complexity implies uniform convergence type guarantees over the entire class of functions.

In particular, this perspective applies to any Lipschitz loss function, whereas our analysis in the main text was specific to the squared loss. We define the data-dependent Rademacher complexity $Rad_{n,\mu}(F)$ by,

$$\mathrm{Rad}_{n,\mu}(\mathcal{F}) = \frac{1}{n}\mathbb{E}^{\sigma_i,x_i}\left[\sup_{f\in\mathcal{F}}\left|\sum_{i=1}^{n}\sigma_i f(x_i)\right|\right] \tag{22}$$

where the values $(\sigma_i)_{i\in[n]}$ are i.i.d. symmetric Rademacher variables in $\{-1, 1\}$ while the values $(x_i)_{i\in[n]}$ are i.i.d. samples from μ.

Suppose $\mu = \Sigma^k_{i=1}\ \alpha_i\mu_i$ is a mixture of *c-isoperimetric* distributions. For finite F consisting of L-Lipschitz f with $|f(x)| \leq 1$ for all $(f, x) \in F \times R^d$, we have

$$Rad_{n,\mu}(\mathcal{F}) \leq O\left(\max\left(\sqrt{\frac{k}{n}}, L\sqrt{\frac{c\log(|\mathcal{F}|)}{nd}}\right)\right) \tag{23}$$

$Rad_{n,\mu}(F)$ measures the ability of functions in F to correlate with random noise. (23) then implies the generalization bound:

For any loss function $l(t, y)$ which is bounded and 1-Lipschitz in its first argument and any $\delta\in[0,1]$, in the setting of (23) we have with probability at least $1-\delta$ the uniform convergence bound:

$$\sup_{f\in\mathcal{F}}\left|\mathbb{E}^{(x,y)\sim\mu}[\ell(f(x),y)] - \frac{1}{n}\sum_{i=1}^{n}\ell(f(x_i),y_i)\right| \leq O\left(\max\left(\sqrt{\frac{k}{n}}, L\sqrt{\frac{c\log(|\mathcal{F}|)}{nd}}, \sqrt{\frac{\log(1/\delta)}{n}}\right)\right)$$

It is important to note that in the classical regime F has rademacher complexity R_F,

$$R_F \leq \sqrt{\frac{\log|F|}{n}}\ \textit{which roughly equates to}\ \sqrt{\frac{P}{n}}$$

And according to our theorem, for Lipschitz function classes F and isoperimetric distributions,

$$R_F \leq \sqrt{\frac{P}{nd}}$$

8. Speculative implications on the real data

We recall from section 1.2 that the MNIST dataset has roughly $n \simeq 10^5$ data points in dimension $d \simeq 10^3$ and [8] showed that the model starts to fit the data robustly i.e. the smooth models with accuracy below the noise level are attained at $p \simeq 2\times10^5$ to 3×10^6. According to theorem 5 law of robustness, $p \simeq 10^{5+3} \simeq 10^8$ should be the required parameter count before the model start to learn robustly. It contradicts the law of robustness since $10^6 \ll 10^8$. But MNIST is not truly 10^3 dimensional. We estimate the true dimension from the law as $d_{eff} \simeq 10^1$.

Now we try to speculate the number of parameters one would require to fit the ImageNet dataset robustly whose $n=1.4\times10^7$ and $d=2\times10^5$. Then assuming the effective dimension d_{eff} scales linearly between the two datasets, we can speculate the effective dimension of the imagenet dataset d_{eff} as 10^3. Using the law of robustness, we get the number of parameters $p \simeq 10^{10}$ to 10^{11}. Current imagenet models are of the order 10^8 to 10^9 parameters. Perhaps it could be the case that robust models for ImageNet do not exist yet because we are not training big enough models on the dataset to be able to fit the data smoothly.

9. Conclusion and Future work

The universal law of robustness shows that one needs to over parameterize approximately by a factor of d i.e $p \simeq n.d$ to be able to train smooth(lipschitz) functions with low training error. The law shows that the trade-off between the size and smoothness is real under the assumptions made in theorem 5.

There are many directions one could take to take the law further. It could be interesting to achieve a more generalized and unrestrictive law of robustness for 1-hidden layer neural networks. The synthetic addition of noise tries to simulate adversaries pertaining to the real world data. Can the law be proven with a more realistic noise assumption? More refined versions of robustness could be explored, with other norms such as l_∞ which is commonly referred to when studying the adversarial robustness of a given network, instead of the euclidean norm. Other norms such as sobolev norms do not work with the universal law of robustness in their current form. Entirely different norms and notions of robustness could exist. What is the scope of law of robustness for specific function classes like the one discussed in section 5.3 for quadratic activation functions, and can one get better bounds on the lipschitz constant?

More importantly, empirical studies that correlate different models trained on different datasets with their robustness in context of the universal law of robustness could help prove the law empirically and gauge how the phenomenon changes in practice. The law could also be extended further to try and find a stronger measure of robustness for generalization.

References

[1] Sebastien Bubeck, Ronen Eldan, Yin Tat Lee, and Dan Mikulincer. Network size and size of the weights in memorization with two-layers neural networks. In Advances in Neural Information Processing Systems, volume 33, pages 4977–4986, 2020.

[2] Sebastien Bubeck, Yuanzhi Li, and Dheeraj M Nagaraj. A law of robustness for two- layer neural networks. In Conference on Learning Theory, pages 804–820. PMLR, 2021.

[3] Sebastien Bubeck, Mark Sellke. A universal law of robustness. arXiv preprint arXiv:2105.12806, 2021.

[4] Eric B Baum. On the capabilities of multilayer perceptrons. Journal of complexity, 4(3):193–215, 1988.

[5] James Alexander and Andre Hirschowitz. Polynomial interpolation in several variables. ´ Journal of Algebraic Geometry, 4(2):201–222, 1995.

[6] Guy Bresler and Dheeraj Nagaraj. A corrective view of neural networks: Representation, memorization and learning. arXiv preprint arXiv:2002.00274, 2020.

[7] George Cybenko. Approximation by superpositions of a sigmoidal function. Mathematics of control, signals and systems, 2(4):303–314, 1989

[8] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In International Conference on Learning Representations, 2018

[9] Moshe Leshno, Vladimir Ya Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. Neural networks, 6(6):861–867, 1993.

[10] Chulhee Yun, Suvrit Sra, and Ali Jadbabaie. Small relu networks are powerful memorizers: a tight analysis of memorization capacity. In Advances in Neural Information Processing Systems, pages 15532–15543, 2019.

[11] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. In International Conference on Learning Representations, 2020.

[12] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. Proceedings of the National Academy of Sciences, 116(32):15849–15854, 2019.

[13] Song Mei and Andrea Montanari. The generalization error of random features regression: Precise asymptotics and the double descent curve. Communications on Pure and Applied Mathematics, 2019.

[14] Peter L. Bartlett, Philip M. Long, Gabor Lugosi, and Alexander Tsigler. Benign ´ overfitting in linear regression. Proceedings of the National Academy of Sciences, 117(48):30063–30070, 2020.