Exploring Generalization in Deep Learning

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, Nathan Srebro
Toyota Technological Institute at Chicago
{bneyshabur, srinadh, mcallester, nati}@ttic.edu

Abstract

With a goal of understanding what drives generalization in deep networks, we consider several recently suggested explanations, including norm-based control, sharpness and robustness. We study how these measures can ensure generalization, highlighting the importance of scale normalization, and making a connection between sharpness and PAC-Bayes theory. We then investigate how well the measures explain different observed phenomena.

1 Introduction

Learning with deep neural networks has enjoyed huge empirical success in recent years across a wide variety of tasks. Despite being a complex, non-convex optimization problem, simple methods such as stochastic gradient descent (SGD) are able to recover good solutions that minimize the training error. More surprisingly, the networks learned this way exhibit good generalization behavior, even when the number of parameters is significantly larger than the amount of training data [18, 28].

In such an over parametrized setting, the objective has multiple global minima, all minimize the training error, but many of them do not generalize well. Hence, just minimizing the training error is not sufficient for learning: picking the wrong global minima can lead to bad generalization behavior. In such situations, generalization behavior depends implicitly on the algorithm used to minimize the training error. Different algorithmic choices for optimization such as the initialization, update rules, learning rate, and stopping condition, will lead to different global minima with different generalization behavior [6, 11, 16]. For example, Neyshabur et al. [16] introduced Path-SGD, an optimization algorithm that is invariant to rescaling of weights, and showed better generalization behavior over SGD for both feedforward and recurrent neural networks [16, 20]. Keskar et al. [11] noticed that the solutions found by stochastic gradient descent with large batch sizes generalizes worse than the one found with smaller batch sizes, and Hardt et al. [9] discuss how stochastic gradient descent ensures uniform stability, thereby helping generalization for convex objectives.

What is the bias introduced by these algorithmic choices for neural networks? What ensures generalization in neural networks? What is the relevant notion of complexity or capacity control?

As mentioned above, simply accounting for complexity in terms of the number of parameters, or any measure which is uniform across all functions representable by a given architecture, is not sufficient to explain the generalization ability of neural networks trained in practice. For linear models, norms and margin-based measures, and not the number of parameters, are commonly used for capacity control [4, 8, 23]. Also norms such as the trace norm and max norm are considered as sensible inductive biases in matrix factorization and are often more appropriate than parameter-counting measures such as the rank [25, 26]. In a similar spirit, Bartlett [2] and later Neyshabur et al. [18] suggested different norms of network parameters to measure the capacity of neural networks. In a different line of work, Keskar et al. [11] suggested "sharpness" (robustness of the training error to perturbations in the parameters) as a complexity measure for neural networks.

What makes a complexity measure appropriate for explaining generalization in deep learning? First, an appropriate complexity measure must be sufficient in ensuring generalization. Second,

networks learned in practice should be of low complexity under this measure. This can happen if our optimization algorithms bias us toward lower complexity models under this measure *and* it is possible to capture real data using networks of low complexity. In particular, the complexity measure should help explain several recently observed empirical phenomena that are not explained by a uniform notion of complexity:

- It is possible to obtain zero training error on random labels using the same architecture for which training with real labels leads to good generalization [28]. We would expect the networks learned using real labels (and which generalizes well) to have much lower complexity, under the suggested measure, than those learned using random labels (and which obviously do not generalize well).
- Increasing the number of hidden units, thereby increasing the number of parameters, can lead to a decrease in generalization error even when the training error does not decrease [18]. We would expect to see the complexity measure decrease as we increase the number of hidden units.
- When training the same architecture, with the same training set, using two different optimization methods (or different algorithmic or parameter choices), one method results in better generalization even though both lead to zero training error [16, 11]. We would expect to see a correlation between the complexity measure and generalization ability among zero-training error models.

In this paper we examine different complexity measures that have recently been suggested, or could be considered, in explaining generalization in deep learning. In light of the above, we evaluate the measures based on their ability to theoretically guarantee generalization, and their empirical ability to explain the above phenomena. Studying how each measure can guarantee generalization also let us better understand how it should be computed and compared when trying to explain the empirical phenomena.

We investigate complexity measures including norms, robustness and sharpness of the network. We emphasize in our theoretical and empirical study the importance of relating the scale of the parameters and the scale of the output of the network, e.g. by relating norm and margin. In this light, we discuss how sharpness by itself is not sufficient for ensuring generalization, but can be combined, through PAC-Bayes analysis, with the norm of the weights to obtain an appropriate complexity measure. In order to further understand the significance of sharpness in deep learning, and how its relationship to margin deviates for that found in linear models, we also establish, in Section 4, sufficient conditions on the network that *provably* ensures small sharpness.

Notation

Let $f_{\mathbf{w}}(\mathbf{x})$ be the function computed by a d layer feed-forward network with parameters $\mathbf{w}$ and Rectified Linear Unit (ReLU) activations, $f_{\mathbf{w}}(\mathbf{x}) = W_d\,\phi(W_{d-1}\,\phi(....\phi(W_1\mathbf{x})))$ where $\phi(z) = \max\{0, z\}$. For a given $\mathbf{x} \in \mathbb{R}^n$, let $D_i^{\mathbf{x},\mathbf{w}}$ denote the diagonal $\{1, 0\}$ matrix corresponding to activation in layer i. To simplify the presentation we drop the $\mathbf{x}$ superscript and use D_i instead. We can therefore write $f_{\mathbf{w}}(\mathbf{x}) = W_d\,D_{d-1}\,W_{d-1}\,\cdots\,D_1\,W_1\,\mathbf{x} = W_d\left(\Pi_{i=1}^{d-1} D_i W_i\right)\mathbf{x}$ where we drop the $\mathbf{x}, \mathbf{w}$ superscript from $D_i^{\mathbf{x},\mathbf{w}}$ and use D_i instead but remember that D_i depends on $\mathbf{x}$ and the parameters W_j for any $j \leq i$.

Let h_i be the number of nodes in layer i, with $h_0 = n$. Therefore, for any layer i, we have $W_i \in R^{h_i \times h_{i-1}}$. Given any input x, the loss of the prediction by the function $f_{\mathbf{w}}$ is then given by $\ell(\mathbf{w}, \mathbf{x})$. We also denote by $L(\mathbf{w})$ the expected loss and by $\widehat{L}(\mathbf{w})$ the empirical loss over the training set. For any integer k, $[k]$ denotes the set $\{1, 2, \cdots, k\}$. Finally, $\|.\|_F$, $\|.\|_2$, $\|.\|_1$, $\|.\|_\infty$ denote Frobenius norm, the spectral norm, element-wise ℓ_1-norm and element-wise ℓ_∞ norm respectively.

2 Generalization and Capacity Control in Deep Learning

In this section, we discuss complexity measures that have been suggested, or could be used for capacity control in neural networks. We discuss advantages and weaknesses of each of these complexity measures and examine their abilities to explain the observed generalization phenomena in deep learning.

We consider the statistical *capacity* of a model class in terms of the number of examples required to ensure *generalization*, i.e. that the population (or test error) is close to the training error, even when minimizing the training error. This also roughly corresponds to the maximum number of examples on which one can obtain small training error even with random labels.

Given a model class $\mathcal{H}$, such as all the functions representable by some feedforward or convolutional networks, one can consider the capacity of the entire class $\mathcal{H}$—this corresponds to learning with a uniform "prior" or notion of complexity over all models in the class. Alternatively, we can also consider some *complexity measure*, which we take as a mapping that assigns a non non-negative number to every hypothesis in the class - $\mathcal{M} : \{\mathcal{H}, S\} \to \mathbb{R}^+$, where S is the training set. It is then sufficient to consider the capacity of the restricted class $\mathcal{H}_{\mathcal{M},\alpha} = \{h : h \in \mathcal{H}, \mathcal{M}(h) \leq \alpha\}$ for a given $\alpha \geq 0$. One can then ensure generalization of a learned hypothesis h in terms of the capacity of $\mathcal{H}_{\mathcal{M},\mathcal{M}(h)}$. Having a good hypothesis with low complexity, and being biased toward low complexity (in terms of $\mathcal{M}$) can then be sufficient for learning, even if the capacity of the entire $\mathcal{H}$ is high. And if we are indeed relying on $\mathcal{M}$ for ensuring generalization (and in particular, biasing toward models with lower complexity under $\mathcal{M}$), we would expect a learned h with lower value of $\mathcal{M}(h)$ to generalize better.

For some of the measures discussed, we allow $\mathcal{M}$ to depend also on the training set. If this is done carefully, we can still ensure generalization for the restricted class $\mathcal{H}_{\mathcal{M},\alpha}$.

We will consider several possible complexity measures. For each candidate measure, we first investigate whether it is sufficient for generalization, and analyze the capacity of $\mathcal{H}_{\mathcal{M},\alpha}$. Understanding the capacity corresponding to different complexity measures also allows us to relate between different measures and provides guidance as to what and how we should measure: From the above discussion, it is clear that any monotone transformation of a complexity measures leads to an equivalent notion of complexity. Furthermore, complexity is meaningful only in the context of a specific hypothesis class $\mathcal{H}$, e.g. specific architecture or network size. The capacity, as we consider it (in units of sample complexity), provides a yardstick by which to measure complexity (we should be clear though, that we are vague regarding the scaling of the generalization error itself, and only consider the scaling in terms of complexity and model class, thus we obtain only a very crude yardstick sufficient for investigating trends and relative phenomena, not a quantitative yardstick).

2.1 Network Size

For any model, if its parameters have finite precision, its capacity is linear in the total number of parameters. Even without making an assumption on the precision of parameters, the VC dimension of feedforward networks can be bounded in terms of the number of parameters $\dim(\mathbf{w})$[1, 2, 5, 21]. In particular, Bartlett [3] and Harvey et al. [10], following Bartlett et al. [5], give the following tight (up to logarithmic factors) bound on the VC dimension and hence capacity of feedforward networks with ReLU activations:

$$\text{VC-dim} = \tilde{O}(d * \dim(\mathbf{w})) \tag{1}$$

In the over-parametrized settings, where the number of parameters is more than the number of samples, complexity measures that depend on the total number of parameters are too weak and cannot explain the generalization behavior. Neural networks used in practice often have significantly more parameters than samples, and indeed can perfectly fit even random labels, obviously without generalizing [28]. Moreover, measuring complexity in terms of number of parameters cannot explain the reduction in generalization error as the number of hidden units increase [18] (see also Figure Figure 4).

2.2 Norms and Margins

Capacity of linear predictors can be controlled independent of the number of parameters, e.g. through regularization of its ℓ_2 norm . Similar norm based complexity measures have also been established for feedforward neural networks with ReLU activations. For example, capacity can be bounded based on the ℓ_1 norm of the weights of hidden units in each layer, and is proportional to $\prod_{i=1}^{d} \|W_i\|_{1,\infty}^2$, where $\|W_i\|_{1,\infty}$ is the maximum over hidden units in layer i of the ℓ_1 norm of incoming weights to the hidden unit [4]. More generally Neyshabur et al. [17] considered group norms $\ell_{p,q}$ corresponding to ℓ_q norm over hidden units of ℓ_p norm of incoming weights to the hidden unit. This includes $\ell_{2,2}$ which is equivalent to the Frobenius norm where the capacity of the network is proportional to

$\prod_{i=1}^{d} \|W_i\|_F^2$. They further motivated a complexity measure that is invariant to node-wise rescaling reparametrization [1], suggesting ℓ_p path norms which is the minimum over all node-wise rescalings of $\prod_{i=1}^{d} \|W_i\|_{p,\infty}$ and is equal to ℓ_p norm of a vector with coordinates each of which is the product of weights along a path from an input node to an output node in the network.

Capacity control in terms of norm, when using a zero/one loss (i.e. counting errors) requires us in addition to account for scaling of the output of the neural networks, as the loss is insensitive to this scaling but the norm only makes sense in the context of such scaling. For example, dividing all the weights by the same number will scale down the output of the network but does not change the $0/1$ loss, and hence it is possible to get a network with arbitrary small norm and the same $0/1$ loss. Using a scale sensitive losses, such as the cross entropy loss, does address this issue (if the outputs are scaled down toward zero, the loss becomes trivially bad), and one can obtain generalization guarantees in terms of norm and the cross entropy loss.

However, we should be careful when comparing the norms of different models learned by minimizing the cross entropy loss, in particular when the training error goes to zero. When the training error goes to zero, in order to push the cross entropy loss (or any other positive loss that diminish at infinity) to zero, the outputs of the network must go to infinity, and thus the norm of the weights (under any norm) should also go to infinity. This means that minimizing the cross entropy loss will drive the norm toward infinity. In practice, the search is terminated at some finite time, resulting in large, but finite norm. But the value of this norm is mostly an indication of how far the optimization is allowed to progress—using a stricter stopping criteria (or higher allowed number of iterations) would yield higher norm. In particular, comparing the norms of models found using different optimization approaches is meaningless, as they would all go toward infinity.

Instead, to meaningfully compare norms of the network, we should explicitly take into account the scaling of the outputs of the network. One way this can be done, when the training error is indeed zero, is to consider the "margin" of the predictions in addition to the norms of the parameters. We refer to the margin for a single data point x as the difference between the score of the correct label and the maximum score of other labels, i.e.

$$f_{\mathbf{w}}(\mathbf{x})[y_{\text{true}}] - \max_{y \neq y_{\text{true}}} f_{\mathbf{w}}(\mathbf{x})[y] \tag{2}$$

In order to measure scale over an entire training set, one simple approach is to consider the "hard margin", which is the minimum margin among all training points. However, this definition is very sensitive to extreme points as well as to the size of the training set. We consider instead a more robust notion that allows a small portion of data points to violate the margin. For a given training set and small value $\epsilon > 0$, we define the margin γ_{margin} as the lowest value of γ such that $\lceil \epsilon m \rceil$ data point have margin lower than γ where m is the size of the training set. We found empirically that the qualitative and relative nature of out empirical results is almost unaffected by reasonable choices of ϵ (e.g. between 0.001 and 0.1).

The norm-based measures we investigate in this work and their corresponding capacity bounds are as follows [2]:

- ℓ_2 norm with capacity proportional to $\frac{1}{\gamma_{\text{margin}}^2} \prod_{i=1}^{d} 4 \|W_i\|_F^2$ [17].
- ℓ_1-path norm with capacity proportional to $\frac{1}{\gamma_{\text{margin}}^2} \left(\sum_{j \in \prod_{k=0}^{d}[h_k]} \left| \prod_{i=1}^{d} 2W_i[j_i, j_{i-1}] \right| \right)^2$ [4, 17].
- ℓ_2-path norm with capacity proportional to $\frac{1}{\gamma_{\text{margin}}^2} \sum_{j \in \prod_{k=0}^{d}[h_k]} \prod_{i=1}^{d} 4h_i W_i^2[j_i, j_{i-1}]$.
- spectral norm with capacity proportional to $\frac{1}{\gamma_{\text{margin}}^2} \prod_{i=1}^{d} h_i \|W_i\|_2^2$.

[1] Node-rescaling can be defined as a sequence of reparametrizations, each of which corresponds to multiplying incoming weights and dividing outgoing weights of a hidden unit by a positive scalar α. The resulting network computes the same function as the network before the reparametrization.

[2] We have dropped the term that only depend on the norm of the input. The bounds based on ℓ_2-path norm and spectral norm can be derived directly from the those based on ℓ_1-path norm and ℓ_2 norm respectively. Without further conditions on weights, exponential dependence on depth is tight but the 4^d dependence might be loose [17]. We will also discuss a rather loose bound on the capacity based on the spectral norm in Section 2.3.

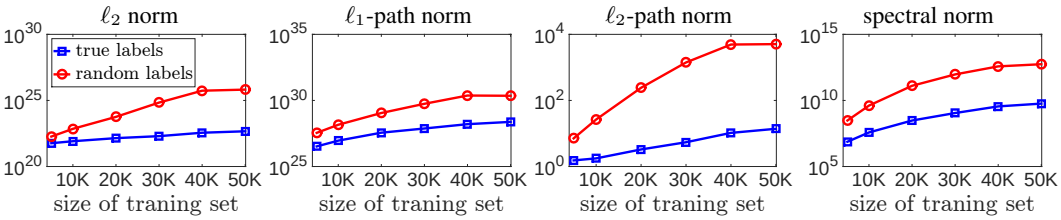

Figure 1: Comparing different complexity measures on a VGG network trained on subsets of CIFAR10 dataset with true (blue line) or random (red line) labels. We plot norm divided by margin to avoid scaling issues (see Section 2), where for each complexity measure, we drop the terms that only depend on depth or number of hidden units; e.g. for ℓ_2-path norm we plot $\gamma_{\text{margin}}^{-2} \sum_{j \in \prod_{k=0}^{d}[h_k]} \prod_{i=1}^{d} W_i^2[j_i, j_{i-1}]$.We also set the margin over training set S to be 5^{th}-percentile of the margins of the data points in S, i.e. $\text{Prc}_5 \{f_{\mathbf{w}}(x_i)[y_i] - \max_{y \neq y_i} f_{\mathbf{w}}(\mathbf{x})[y] | (x_i, y_i) \in S\}$. In all experiments, the training error of the learned network is zero. The plots indicate that these measures can explain the generalization as the complexity of model learned with random labels is always higher than the one learned with true labels. Furthermore, the gap between the complexity of models learned with true and random labels increases as we increase the size of the training set.

where $\prod_{k=0}^{d}[h_k]$ is the Cartesian product over sets $[h_k]$. The above bounds indicate that capacity can be bounded in terms of either ℓ_2-norm or ℓ_1-path norm independent of number of parameters. The ℓ_2-path norm dependence on the number of hidden units in each layer is unavoidable. However, it is not clear that the dependence on the number of parameters is needed for the bound based on the spectral norm.

As an initial empirical investigation of the appropriateness of the different complexity measures, we compared the complexity (under each of the above measures) of models trained on true versus random labels. We would expect to see two phenomena: first, the complexity of models trained on true labels should be substantially lower than those trained on random labels, corresponding to their better generalization ability. Second, when training on random labels, we expect capacity to increase almost linearly with the number of training examples, since every extra example requires new capacity in order to fit it's random label. However, when training on true labels we expect the model to capture the true functional dependence between input and output and thus fitting more training examples should only require small increases in the capacity of the network. The results are reported in Figure 1. We indeed observe a gap between the complexity of models learned on real and random labels for all four norms, with the difference in increase in capacity between true and random labels being most pronounced for the ℓ_2 norm and ℓ_2-path norm.

In Section 3 we present further empirical investigations of the appropriateness of these complexity measures to explaining other phenomena.

2.3 Lipschitz Continuity and Robustness

The measures/norms we discussed so far also control the Lipschitz constant of the network with respect to its input. Is the capacity control achieved through the bound on the Lipschitz constant? Is bounding the Lipschitz constant alone enough for generalization? To answer these questions, and in order to understand capacity control in terms of Lipschitz continuity more broadly, we review here the relevant guarantees.

Given an input space $\mathcal{X}$ and metric $\mathcal{M}$, a function $f : \mathcal{X} \to \mathbb{R}$ on a metric space $(\mathcal{X}, \mathcal{M})$ is called a Lipschitz function if there exists a constant $C_{\mathcal{M}}$, such that $|f(x) - f(y)| \leq C_{\mathcal{M}} \mathcal{M}(x, y)$. Luxburg and Bousquet [13] studied the capacity of functions with bounded Lipschitz constant on metric space $(\mathcal{X}, \mathcal{M})$ with a finite diameter $\text{diam}_{\mathcal{M}}(\mathcal{X}) = \sup_{x,y \in X} \mathcal{M}(x, y)$ and showed that the capacity is proportional to $\left(\frac{C_{\mathcal{M}}}{\gamma_{\text{margin}}}\right)^n \text{diam}_{\mathcal{M}}(\mathcal{X})$. This capacity bound is weak as it has an exponential dependence on input size.

Another related approach is through algorithmic robustness as suggested by Xu and Mannor [27]. Given $\epsilon > 0$, the model $f_{\mathbf{w}}$ found by a learning algorithm is K robust if $\mathcal{X}$ can be partitioned into K

disjoint sets, denoted as $\{C_i\}_{i=1}^K$, such that for any pair $(\mathbf{x}, y)$ in the training set $\mathbf{s}$,[3]

$$\mathbf{x}, \mathbf{z} \in C_i \Rightarrow |\ell(\mathbf{w}, \mathbf{x}) - \ell(\mathbf{w}, \mathbf{z})| \leq \epsilon \tag{3}$$

Xu and Mannor [27] showed the capacity of a model class whose models are K-robust scales as K. For the model class of functions with bounded Lipschitz $C_{\|.\|}$, K is proportional to $\frac{C_{\|\cdot\|}}{\gamma_{\text{out}}}$-covering number of the input domain $\mathcal{X}$ under norm $\|.\|$. However, the covering number of the input domain can be exponential in the input dimension and the capacity can still grow as $\left(\frac{C_{\|\cdot\|}}{\gamma_{\text{margin}}}\right)^n$ [4].

Returning to our original question, the C_{ℓ_∞} and C_{ℓ_2} Lipschitz constants of the network can be bounded by $\prod_{i=1}^d \|W_i\|_{1,\infty}$ (hence ℓ_1-path norm) and $\prod_{i=1}^d \|W_i\|_2$, respectively [27, 24]. This will result in a very large capacity bound that scales as $\left(\frac{\prod_{i=1}^d \|W_i\|_2}{\gamma_{\text{margin}}}\right)^n$, which is exponential in both the input dimension and depth of the network. This shows that simply bounding the Lipschitz constant of the network is not enough to get a reasonable capacity control, and the capacity bounds of the previous Section are not merely a consequence of bounding the Lipschitz constant.

2.4 Sharpness

The notion of sharpness as a generalization measure was recently suggested by Keskar et al. [11] and corresponds to robustness to adversarial perturbations on the parameter space:

$$\zeta_\alpha(\mathbf{w}) = \frac{\max_{|\boldsymbol{\nu}| \leq \alpha(|\mathbf{w}|+\mathbf{1})} \widehat{L}(f_{\mathbf{w}+\boldsymbol{\nu}}) - \widehat{L}(f_{\mathbf{w}})}{1 + \widehat{L}(f_{\mathbf{w}})} \simeq \max_{|\boldsymbol{\nu}| \leq \alpha(|\mathbf{w}|+\mathbf{1})} \widehat{L}(f_{\mathbf{w}+\boldsymbol{\nu}}) - \widehat{L}(f_{\mathbf{w}}), \tag{4}$$

where the training error $\widehat{L}(f_{\mathbf{w}})$ is generally very small in the case of neural networks in practice, so we can simply drop it from the denominator without a significant change in the sharpness value.

As we will explain below, sharpness defined this way does *not* capture the generalization behavior. To see this, we first examine whether sharpness can predict the generalization behavior for networks trained on true vs random labels. In the left plot of Figure 2, we plot the sharpness for networks trained on true vs random labels. While sharpness correctly predicts the generalization behavior for bigger networks, for networks of smaller size, those trained on random labels have less sharpness than the ones trained on true labels. Furthermore sharpness defined above depends on the scale of $\mathbf{w}$ and can be artificially increased or decreased by changing the scale of the parameters. Therefore, sharpness alone is not suffficient to control the capacity of the network.

Instead, we propose viewing a related notion of expected sharpness in the context of the PAC-Bayesian framework. Viewed this way, it becomes clear that sharpness controls only one of two relevant terms, and must be balanced with some other measure such as norm. Together, sharpness and norm do provide capacity control and can explain many of the observed phenomena.

The PAC-Bayesian framework [14, 15] provides guarantees on the expected error of a randomized predictor (hypothesis), drawn form a distribution denoted $\mathcal{Q}$ and sometimes referred to as a "posterior" (although it need *not* be the Bayesian posterior), that depends on the training data. In our case, we consider a distribution $\mathcal{Q}$ over networks with weights of the form $\mathbf{w} + \boldsymbol{\nu}$, where $\mathbf{w}$ is a single predictor learned from the training set, and $\boldsymbol{\nu}$ is a random variable. Then, given a "prior" distribution P over the hypothesis that is independent of the training data, with probability at least $1 - \delta$ over the draw of the training data, the expected error of $f_{\mathbf{w}+\boldsymbol{\nu}}$ can be bounded as follows:

$$\mathbb{E}_{\boldsymbol{\nu}}[L(f_{\mathbf{w}+\boldsymbol{\nu}})] \leq \widehat{L}(f_{\mathbf{w}}) + \underbrace{\mathbb{E}_{\boldsymbol{\nu}}[\widehat{L}(f_{\mathbf{w}+\boldsymbol{\nu}})] - \widehat{L}(f_{\mathbf{w}})}_{\text{expected sharpness}} + \sqrt{\frac{1}{m}\left(KL\left(\mathbf{w} + \boldsymbol{\nu} \| P\right) + \ln(\frac{2m}{\delta})\right)} \tag{5}$$

As we can see, the PAC-Bayes bound depends on two quantities - i) the expected sharpness and ii) the Kullback Leibler (KL) divergence to the "prior" P. The bound is valid for any distribution measure P, any perturbation distribution $\boldsymbol{\nu}$ and any method of choosing $\mathbf{w}$ from the training set. A simple way to instantiate the bound is to set P to be a zero mean, unit variance Gaussian distribution. Choosing

[3] Xu and Mannor [27] have defined the robustness as a property of learning algorithm given the model class and the training set. Here since we are focused on the learned model, we introduce it as a property of the model.

[4] Similar to margin-based bounds, we drop the term that depends on the diameter of the input space.

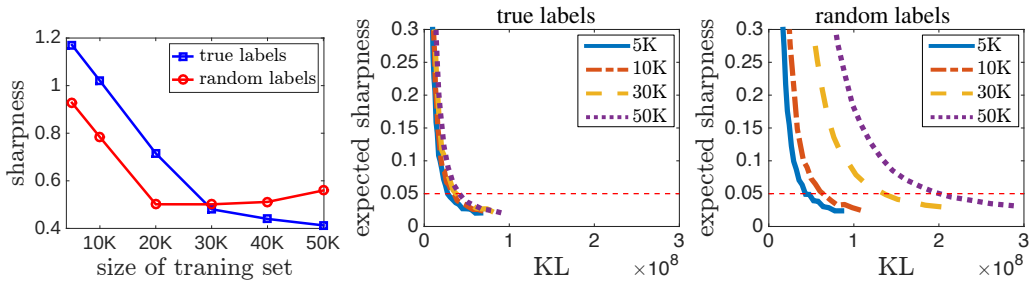

Figure 2: Sharpness and PAC-Bayes measures on a VGG network trained on subsets of CIFAR10 dataset with true or random labels. In the left panel, we plot max sharpness, which we calculate as suggested by Keskar et al. [11] where the perturbation for parameter w_i has magnitude $5.10^{-4}(|w_i| + 1)$. The middle and right plots demonstrate the relationship between expected sharpness and KL divergence in PAC-Bayes analysis for true and random labels respectively. For PAC-Bayes plots, each point in the plot correspond to a choice of variable α where the standard deviation of the perturbation for the parameter i is $\alpha(10\,|w_i| + 1)$. The corresponding KL to each α is nothing but weighted ℓ_2 norm where the weight for each parameter is the inverse of the standard deviation of the perturbation.

the perturbation $\boldsymbol{\nu}$ to also be a zero mean spectral Gaussian with variance σ^2 in every direction, yields the following guarantee (w.p. $1 - \delta$ over the training set):

$$\mathbb{E}_{\boldsymbol{\nu}\sim\mathcal{N}(0,\sigma)^n}[L(f_{\mathbf{w}+\boldsymbol{\nu}})] \leq \widehat{L}(f_{\mathbf{w}}) + \underbrace{\mathbb{E}_{\boldsymbol{\nu}\sim\mathcal{N}(0,\sigma)^n}[\widehat{L}(f_{\mathbf{w}+\boldsymbol{\nu}})] - \widehat{L}(f_{\mathbf{w}})}_{\text{expected sharpness}} + \sqrt{\frac{1}{m}\left(\underbrace{\frac{\|\mathbf{w}\|_2^2}{2\sigma^2}}_{\text{KL}} + \ln\frac{1}{\delta}\right)}, \quad (6)$$

Another interesting approach is to set the variance of the perturbation to each parameter with respect to the magnitude of the parameter. For example if $\sigma_i = \alpha\,|w_i| + \beta$, then the KL term in the above expression changes to $\sum_i \frac{w_i^2}{2\sigma_i^2}$.

The above generalization guarantees give a clear way to think about capacity control jointly in terms of both the expected sharpness and the norm, and as we discussed earlier indicates that sharpness by itself cannot control the capacity without considering the scaling. In the above generalization bound, norms and sharpness interact in a direct way depending on σ, as increasing the norm by decreasing σ causes decrease in sharpness and vice versa. It is therefore important to find the right balance between the norm and sharpness by choosing σ appropriately in order to get a reasonable bound on the capacity.

In our experiments we observe that looking at both these measures jointly indeed makes a better predictor for the generalization error. Since the precise way the sharpness and KL-divergence are combined in (6) is not tight, we do not consider this specific bound. Instead, we propose using bi-criteria plots, where sharpness and KL-divergence are plotted against each other, as we vary the perturbation variance. For example, in the center and right panels of Figure 2 we show such plots for networks trained on true and random labels respectively. We see that although sharpness by itself is not sufficient for explaining generalization in this setting (as we saw in the left panel), the bi-criteria plots are significantly lower for the true labels. Even more so, the change in the bi-criteria plot as we increase the number of samples is significantly larger with random labels, correctly capturing the required increase in capacity. For example, to get a fixed value of expected sharpness such as $\epsilon = 0.05$, networks trained with random labels require higher norm compared to those trained with true labels. This behavior is in agreement with our earlier discussion, that sharpness is sensitive to scaling of the parameters and is not a capacity control measure as it can be artificially changed by scaling the network. However, combined with the norm, sharpness does seem to provide a capacity measure.

This connection between sharpness and the PAC-Bayesian framework was also recently noticed by Dziugaite and Roy [7], who optimize the PAC-Bayes generalization bound over a family of multivariate Gaussian distributions, extending the work of Langford and Caruana [12]. They show that the optimized PAC-Bayes bounds are numerically non-vacuous for feedforward networks trained on a binary classification variant of MNIST dataset.

3 Empirical Investigation

In this section we investigate the ability of the discussed measures to explain the the generalization phenomenon discussed in the Introduction. We already saw in Figures 1 and 2 that these measures capture the difference in generalization behavior of models trained on true or random labels, including the increase in capacity as the sample size increases, and the difference in this increase between true and random labels.

Different Global Minima

Given different global minima of the training loss on the same training set and with the same model class, can these measures indicate which model is going to generalize better? In order to verify this property, we can calculate each measure on several different global minima and see if lower values of the measure imply lower generalization error. In order to find different global minima for the training loss, we design an experiment where we force the optimization methods to converge to different global minima with varying generalization abilities by forming a confusion set that includes samples with random labels. The optimization is done on the loss that includes examples from both the confusion set and the training set. Since deep learning models have very high capacity, the optimization over the union of confusion set and training set generally leads to a point with zero error over both confusion and training sets which thus is a global minima for the training set.

We randomly select a subset of CIFAR10 dataset with 10000 data points as the training set and our goal is to find networks that have zero error on this set but different generalization abilities on the test set. In order to do that, we train networks on the union of the training set with fixed size 10000 and confusion sets with varying sizes that consists of CIFAR10 samples with random labels; and we evaluate the learned model on an independent test set. The trained network achieves zero training error but as shown in Figure 3, the test error of the model increases with increasing size of the confusion set. The middle panel of this Figure suggests that the norm of the learned networks can indeed be predictive of their generalization behavior. However, we again observe that sharpness has a poor behavior in these experiments. The right panel of this figure also suggests that PAC-Bayes measure of joint sharpness and KL divergence, has better behavior - for a fixed expected sharpness, networks that have higher generalization error, have higher norms.

Increasing Network Size

We also repeat the experiments conducted by Neyshabur et al. [18] where a fully connected feedforward network is trained on MNIST dataset with varying number of hidden units and we check the values of different complexity measures on each of the learned networks.The left panel in Figure 4 shows the training and test error for this experiment. While 32 hidden units are enough to fit the training data, we observe that networks with more hidden units generalize better. Since the optimization is done without any explicit regularization, the only possible explanation for this phenomenon is the implicit regularization by the optimization algorithm. Therefore, we expect a sensible complexity measure to decrease beyond 32 hidden units and behave similar to the test error. Different measures are reported for learned networks. The middle panel suggest that all margin/norm based complexity measures decrease for larger networks up to 128 hidden units. For networks with more hidden units, ℓ_2 norm and ℓ_1-path norm increase with the size of the network. The middle panel suggest that ℓ_2-path norm can provide some explanation for this phenomenon. However, as we discussed in Section 2, the actual complexity measure based on ℓ_2-path norm also depends on the number of hidden units and taking this into account indicates that the measure based on ℓ_2-path norm cannot explain this phenomenon. This is also the case for the margin based measure that depends on the spectral norm. In subsection 2.3 we discussed another complexity measure that also depends the spectral norm through Lipschitz continuity or robustness argument. Even though this bound is very loose, it is monotonic with respect to the spectral norm that is reported in the plots. Unfortunately, we do observe some increase in spectral norm by increasing number of hidden units beyond 512. The right panel shows that the joint PAC-Bayes measure decrease for larger networks up to size 128 but fails to explain this generalization behavior for larger ntworks. This suggests that the measures looked so far are not sufficient to explain all the generalization phenomenon observed in neural networks.

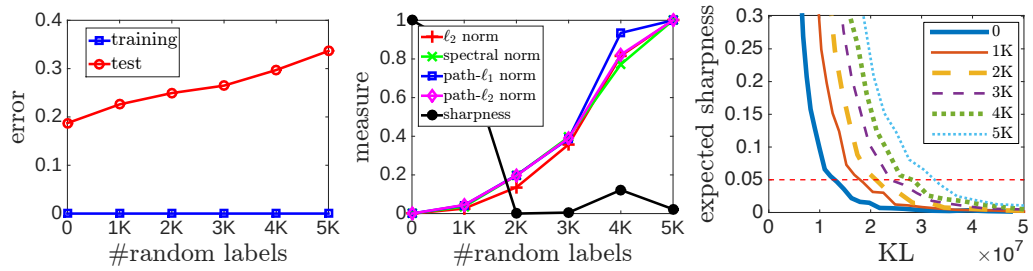

Figure 3: Experiments on global minima with poor generalization. For each experiment, a VGG network is trained on union of a subset of CIFAR10 dataset with size 10000 containing samples with true labels and another subset of CIFAR10 datasets with varying size containing random labels. The learned networks are all global minima for the objective function on the subset with true labels. The left plot indicates the training and test errors based on the size of the set with random labels. The plot in the middle shows change in different measures based on the size of the set with random labels. The plot on the right indicates the relationship between expected sharpness and KL in PAC-bayes for each of the experiments. Measures are calculated as explained in Figures 1 and 2.

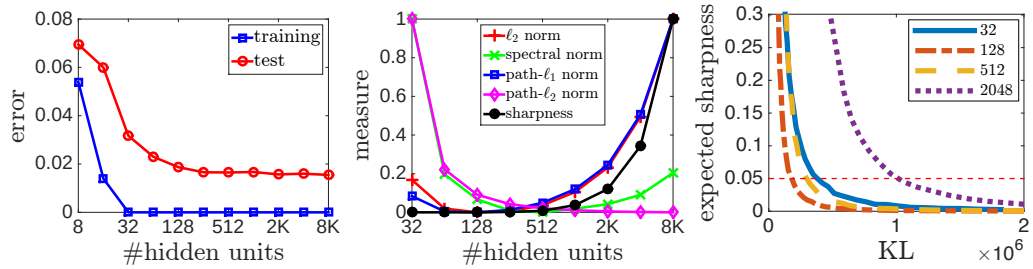

Figure 4: The generalization of two layer perceptron trained on MNIST dataset with varying number of hidden units. The left plot indicates the training and test errors. The test error decreases as the size increases. The middle plot shows different measures for each of the trained networks. The plot on the right indicates the relationship between expected sharpness and KL in PAC-Bayes for each of the experiments. Measures are calculated as explained in Figures 1 and 2.

4 Bounding Sharpness

So far we have discussed margin based and sharpness based complexity measures to understand capacity. We have also discussed how sharpness based complexity measures in combination with norms characterize the generalization behavior under the PAC-Bayes framework. In this section we study the question of what affects the sharpness of neural networks? For the case of linear predictors, sharpness only depends on the norm of the predictor. In contrast, for multilayered networks, interaction between the layers plays a major role and consequently two different networks with the same norm can have drastically different sharpness values. For example, consider a network where some subset of the layers despite having non-zero norm interact weakly with their neighbors, or are almost orthogonal to each other. Such a network will have very high sharpness value compared to a network where the neighboring layers interact strongly.

In this section we establish sufficient conditions to bound the expected sharpness of a feedforward network with ReLU activations. Such conditions serve as a useful guideline in studying what helps an optimization method to converge to less sharp optima. Unlike existing generalization bounds [4, 17, 13, 27, 24], our sharpness based bound does not suffer from exponential dependence on depth.

Now we discuss the conditions that affect the sharpness of a network. As discussed earlier, weak interactions between layers can cause the network to have high sharpness value. Condition $C1$ below prevents such weak interactions (cancellations). A network can also have high sharpness if the changes in the number of activations is exponential in the perturbations to its weights, even for small perturbations. Condition $C2$ avoids such extreme situations on activations. Finally, if a non-active node with large weights becomes active because of the perturbations in lower layers, that can lead to

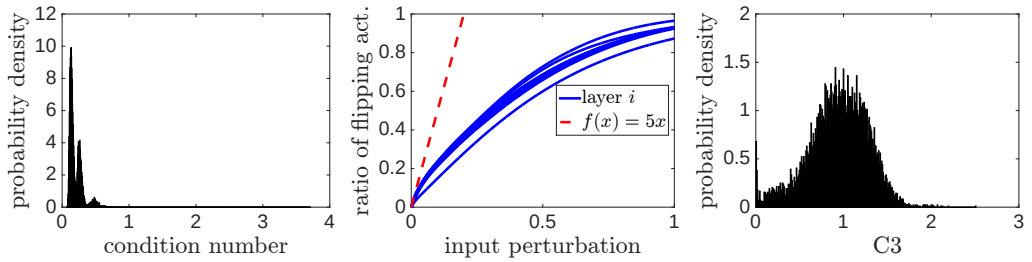

Figure 5: Verifying the conditions of Theorem 1 on a 10 layer perceptron with 1000 hidden units in each layer, i.e. more than 10,000,000 parameters on MNIST. We have numerically checked that all values are within the displayed range. **Left**: $C1$: condition number of the network, i.e. $\frac{1}{\mu}$. **Middle**: $C2$: the ratio of activations that flip based on magnitude of perturbation. **Right**: $C3$: the ratio of norm of incoming weights to each hidden units with respect to average of the same quantity over hidden units in the layer.

huge changes to the output of the network. Condition $C3$ prevents having such spiky (in magnitude) hidden units. This leads us to the following three conditions, that help in avoiding such pathological cases.

$(C1)$: Given x, let $x = W_0$ and $D_0 = I$. Then, for all $0 \leq a < c < b \leq d$, $\| \left(\Pi_{i=a}^{b} D_i W_i\right) \|_F \geq \frac{\mu}{\sqrt{h_c}} \|\Pi_{i=c+1}^{b} D_i W_i\|_F \| \left(\Pi_{i=a}^{c} D_i W_i\right) \|_F$.

$(C2)$: Given x, for any level k, $\frac{1}{h_k} \sum_{i \in [h_k]} 1_{W_{k,i} \Pi_{j=1}^{k-1} D_j W_j x \leq \delta} \leq C_2 \delta$.

$(C3)$: For all i, $\|W_i\|_{2,\infty}^2 h_i \leq C_3^2 \|D_i W_i\|_F^2$.

Here, $W_{k,i}$ denotes the weights of the i^{th} output node in layer k. $\|W_i\|_{2,\infty}$ denotes the maximum $L2$ norm of a hidden unit in layer i. Now we state our result on the generalization error of a ReLU network, in terms of average sharpness and its norm. Let $C_{\delta_i} = (1+\delta_i)\sqrt{2\ln(h_i)}$ and $\|x\| = 1$.

Theorem 1. *Let $\boldsymbol{\nu}_i$ be a random $h_i \times h_{i-1}$ matrix with each entry distributed according to $\mathcal{N}(0, \sigma_i^2)$ and let $\gamma_i = \frac{\sigma_i \sqrt{h_i} \sqrt{h_{i-1}}}{\mu^2 \|W_i\|_F}$. Then, under the conditions $C1, C2, C3$, with probability $\geq 1 - \sum_{i=1}^{d} \delta_i$,*

$$\mathbb{E}_{\boldsymbol{\nu} \sim \mathcal{N}(0,\sigma)^n}[L(f_{\mathbf{w}+\boldsymbol{\nu}})] - \widehat{L}(f_{\mathbf{w}}) \leq O\left(\left[\Pi_{i=1}^{d} (1+\gamma_i) - 1 \right.\right.$$
$$\left.\left. + \Pi_{i=1}^{d} (1+\gamma_i C_2 C_3) \left(\Pi_{i=1}^{d} (1+\gamma_i C_{\delta_i} C_2) - 1 \right) \right] C_L \sum_x \frac{\|f_{\mathbf{w}}(x)\|_F}{m} \right) + \sqrt{\frac{1}{m} \left(\sum_{i=1}^{d} \frac{\|W_i\|_F^2}{\sigma_i^2} + \ln \frac{1}{\delta} \right)}.$$

To understand the above generalization error bound, consider choosing $\gamma_i = \frac{\sigma}{C_{\delta_i} d}$, and we get a bound that simplifies as follows:

$$\mathbb{E}_{\boldsymbol{\nu} \sim \mathcal{N}(0,\sigma)^n}[\widehat{L}(f_{\mathbf{w}+\boldsymbol{\nu}})] - \widehat{L}(f_{\mathbf{w}}) \leq O\left(\sigma \left(1 + (1+\sigma C_2 C_3) C_2\right) C_L \frac{\sum_x \|f_{\mathbf{w}}(x)\|_F}{m} \right)$$
$$+ \sqrt{\frac{1}{m} \left(\frac{d^2}{\mu^4} \sum_{i=1}^{d} \frac{h_i h_{i-1}}{\sigma^2} + \ln \frac{1}{\delta} \right)}$$

If we choose large σ, then the network will have large expected sharpness but smaller 'norm' and vice versa. Now one can optimize over the σ to balance between the terms on the right hand side and get a better capacity bound. For any reasonable choice of σ, the generalization error above, depend only linearly on depth and does not have any exponential dependence, unlike other notions of generalization. Also the error gets worse with decreasing μ and increasing C_2, C_3 as the sharpness of the network increases which is in accordance with our discussion of the conditions above.

Additionally the conditions $C1 - C3$ actually hold for networks trained in practice as we verify in Figure 5, and our experiments suggest that, $\mu \geq 1/4, C2 \leq 5$ and $C3 \leq 3$. More details on the verification and comparing the conditions on learned network with those of random weights, are presented in the appendix.

Proof of Theorem 1 Define $f_{\{W_{-i-j},\boldsymbol{\nu}_{i,j}\}}(x)$ as the network $f_{\mathbf{w}}$ with weights in layers $i, j,, W_i, W_j$ replaced by $\boldsymbol{\nu}_i, \boldsymbol{\nu}_j$. For a given $x \in \mathbb{R}^n$,

$$\begin{aligned}
&\mathbb{E}\left|\widehat{L}(f_{\mathbf{w}+\boldsymbol{\nu}}(x)) - \widehat{L}(f_{\mathbf{w}}(x))\right| \\
&\quad\leq C_L \mathbb{E}\|f_{\mathbf{w}+\boldsymbol{\nu}}(x) - f_{\mathbf{w}}(x)\|_F \\
&\quad\overset{(i)}{=} C_L \mathbb{E}\|(W+\boldsymbol{\nu})_d \left(\Pi_{i=1}^{d-1} \widehat{D}_i (W+\boldsymbol{\nu})_i\right) * x - W_d \left(\Pi_{i=1}^{d-1} D_i W_i\right) * x\|_F \\
&\quad\leq C_L \mathbb{E}\|(W+\boldsymbol{\nu})_d \left(\Pi_{i=1}^{d-1} D_i (W+\boldsymbol{\nu})_i\right) * x - W_d \left(\Pi_{i=1}^{d-1} D_i W_i\right) * x\|_F \\
&\qquad + C_L \mathbb{E}\|(W+\boldsymbol{\nu})_d \left(\Pi_{i=1}^{d-1} \widehat{D}_i (W+\boldsymbol{\nu})_i\right) * x - (W+\boldsymbol{\nu})_d \left(\Pi_{i=1}^{d-1} D_i (W+\boldsymbol{\nu})_i\right) * x\|_F \\
&\quad\leq C_L \mathbb{E}\|(W+\boldsymbol{\nu})_d \left(\Pi_{i=1}^{d-1} D_i (W+\boldsymbol{\nu})_i\right) * x - W_d \left(\Pi_{i=1}^{d-1} D_i W_i\right) * x\|_F + C_L \mathbb{E}\|Err_d\|_F,
\end{aligned} \tag{7}$$

where $Err_d = \|(W+\boldsymbol{\nu})_d \left(\Pi_{i=1}^{d-1} \widehat{D}_i (W+\boldsymbol{\nu})_i\right) * x - (W+\boldsymbol{\nu})_d \left(\Pi_{i=1}^{d-1} D_i (W+\boldsymbol{\nu})_i\right) * x\|_F$. (i) $\widehat{D}_i$ is the diagonal matrix with 0's and 1's corresponding to the activation pattern of the perturbed network $f_{\mathbf{w}+\boldsymbol{\nu}}(x)$.

The first term in the equation (7) corresponds to error due to perturbation of a network with unchanged activations (linear network). Intuitively this is small when any subset of successive layers of the network do no interact weakly with each other (not orthogonal to each other). Condition $C1$ captures this intuition and we bound this error in Lemma 8.

Lemma 1. *Let $\boldsymbol{\nu}_i$ be a random $h_i \times h_{i-1}$ matrix with each entry distributed according to $\mathcal{N}(0, \sigma_i^2)$. Then, under the condition $C1$,*

$$\begin{aligned}
&\mathbb{E}\|(W+\boldsymbol{\nu})_d \left(\Pi_{i=1}^{d-1} D_i (W+\boldsymbol{\nu})_i\right) * x - W_d \left(\Pi_{i=1}^{d-1} D_i W_i\right) * x\|_F \\
&\qquad\qquad \leq \left(\Pi_{i=1}^{d} \left(1 + \frac{\sigma_i \sqrt{h_i}\sqrt{h_{i-1}}}{\mu^2 \|D_i W_i\|_F}\right) - 1\right) \|f_{\mathbf{w}}(x)\|_F.
\end{aligned}$$

The second term in the equation (7) captures the perturbation error due to change in activations. If a tiny perturbation can cause exponentially many changes in number of active nodes, then that network will have huge sharpness. Condition $C2$ and $C3$ essentially characterize the behavior of sensitivity of activation patterns to perturbations, leading to a bound on this term in Lemma 2.

Lemma 2. *Let $\boldsymbol{\nu}_i$ be a random $h_i \times h_{i-1}$ matrix with each entry distributed according to $\mathcal{N}(0, \sigma_i^2)$ and let $\gamma_i = \frac{\sigma_i \sqrt{h_i}\sqrt{h_{i-1}}}{\mu^2 \|W_i\|_F}$. Then, under the conditions $C1$, $C2$ and $C3$, with probability $\geq 1 - \sum_{i=1}^{d} \delta_i$, for all $1 \leq k \leq d$,*

$$\|\widehat{D}_k - D_k\|_1 \leq O\left(C_2 h_k C_{\delta_k} \sigma_k \|f_{\mathbf{w}}^{k-1}\|_F\right)$$

and

$$\mathbb{E}\|Err_k\|_F \leq O\left(\Pi_{i=1}^{k} \left(1 + \frac{\sigma_i \sqrt{h_i}\sqrt{h_{i-1}} C_2 C_3}{\mu^2 \|W_i\|_F}\right) \left(\Pi_{i=1}^{k} (1 + \frac{\sigma_i \sqrt{h_i}\sqrt{h_{i-1}} C_{\delta_i} C_2}{\mu^2 \|W_i\|_F}) - 1\right) \|f_{\mathbf{w}}^{k}\|_F\right).$$

Hence, from Lemma 8 and Lemma 2 we get,

$$\begin{aligned}
&\mathbb{E}\left|\widehat{L}(f_{\mathbf{w}+\boldsymbol{\nu}}(x)) - \widehat{L}(f_{\mathbf{w}}(x))\right| \\
&\leq \left[\Pi_{i=1}^{d} (1+\gamma_i) - 1 + \Pi_{i=1}^{d} (1 + \gamma_i C_2 C_3) \left(\Pi_{i=1}^{d} (1 + \gamma_i C_{\delta_i} C_2) - 1\right)\right] C_L \|f_{\mathbf{w}}(x)\|_F.
\end{aligned}$$

Here $\gamma_i = \frac{\sigma_i \sqrt{h_i}\sqrt{h_{i-1}}}{\mu^2 \|W_i\|_F}$. Substituting the above bound on expected sharpness in the PAC-Bayes result (equation (2.4)), gives the result.

5 Conclusion

Learning with deep neural networks displays good generalization behavior in practice, a phenomenon that remains largely unexplained. In this paper we discussed different candidate complexity measures

that might explain generalization in neural networks. We outline a concrete methodology for investigating such measures, and report on experiments studying how well the measures explain different phenomena. While there is no clear choice yet, some combination of expected sharpness and norms do seem to capture much of the generalization behavior of neural networks. A major issue still left unresolved is how the choice of optimization algorithm biases such complexity to be low, and what is the precise relationship between optimization and implicit regularization.

References

[1] M. Anthony and P. L. Bartlett. *Neural network learning: Theoretical foundations*. cambridge university press, 2009.

[2] P. L. Bartlett. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE transactions on Information Theory*, 44(2):525–536, 1998.

[3] P. L. Bartlett. The impact of the nonlinearity on the VC-dimension of a deep network. *Preprint*, 2017.

[4] P. L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[5] P. L. Bartlett, V. Maiorov, and R. Meir. Almost linear vc dimension bounds for piecewise polynomial networks. *Neural computation*, 10(8):2159–2173, 1998.

[6] P. Chaudhari, A. Choromanska, S. Soatto, and Y. LeCun. Entropy-sgd: Biasing gradient descent into wide valleys. *arXiv preprint arXiv:1611.01838*, 2016.

[7] G. K. Dziugaite and D. M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017.

[8] T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advances in computational mathematics*, 13(1):1–50, 2000.

[9] M. Hardt, B. Recht, and Y. Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *ICML*, 2016.

[10] N. Harvey, C. Liaw, and A. Mehrabian. Nearly-tight vc-dimension bounds for piecewise linear neural networks. *arXiv preprint arXiv:1703.02930*, 2017.

[11] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

[12] J. Langford and R. Caruana. (not) bounding the true error. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, pages 809–816. MIT Press, 2001.

[13] U. v. Luxburg and O. Bousquet. Distance-based classification with lipschitz functions. *Journal of Machine Learning Research*, 5(Jun):669–695, 2004.

[14] D. A. McAllester. Some PAC-Bayesian theorems. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 230–234. ACM, 1998.

[15] D. A. McAllester. PAC-Bayesian model averaging. In *Proceedings of the twelfth annual conference on Computational learning theory*, pages 164–170. ACM, 1999.

[16] B. Neyshabur, R. Salakhutdinov, and N. Srebro. Path-SGD: Path-normalized optimization in deep neural networks. In *Advanced in Neural Information Processsing Systems (NIPS)*, 2015.

[17] B. Neyshabur, R. Tomioka, and N. Srebro. Norm-based capacity control in neural networks. In *Proceeding of the 28th Conference on Learning Theory (COLT)*, 2015.

[18] B. Neyshabur, R. Tomioka, and N. Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *Proceeding of the International Conference on Learning Representations workshop track*, 2015.

[19] B. Neyshabur, R. Tomioka, R. Salakhutdinov, and N. Srebro. Data-dependent path normalization in neural networks. In *the International Conference on Learning Representations*, 2016.

[20] B. Neyshabur, Y. Wu, R. Salakhutdinov, and N. Srebro. Path-normalized optimization of recurrent neural networks with relu activations. *Advances in Neural Information Processing Systems*, 2016.

[21] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[22] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[23] A. J. Smola, B. Schölkopf, and K.-R. Müller. The connection between regularization operators and support vector kernels. *Neural networks*, 11(4):637–649, 1998.

[24] J. Sokolic, R. Giryes, G. Sapiro, and M. R. Rodrigues. Generalization error of invariant classifiers. *arXiv preprint arXiv:1610.04574*, 2016.

[25] N. Srebro and A. Shraibman. Rank, trace-norm and max-norm. In *International Conference on Computational Learning Theory*, pages 545–560. Springer Berlin Heidelberg, 2005.

[26] N. Srebro, J. Rennie, and T. S. Jaakkola. Maximum-margin matrix factorization. In *Advances in neural information processing systems*, pages 1329–1336, 2005.

[27] H. Xu and S. Mannor. Robustness and generalization. *Machine learning*, 86(3):391–423, 2012.

[28] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017.

A Experiments Settings

In experiment with different network sizes, we train a two layer perceptron with ReLU activation and varying number of hidden units without Batch Normalization or dropout. In the rest of the experiments, we train a modified version of the VGG architecture [22] with the configuration $2 \times [64, 3, 3, 1]$, $2 \times [128, 3, 3, 1]$, $2 \times [256, 3, 3, 1]$, $2 \times [512, 3, 3, 1]$ where we add Batch Normalization before ReLU activations and apply 2×2 max-pooling with window size 2 and dropout after each stack. Convolutional layers are followed by 4×4 average pooling, a fully connected layer with 512 hidden units and finally a linear layer is added for prediction.

In all experiments we train the networks using stochastic gradient descent (SGD) with mini-batch size 64, fixed learning rate 0.01 and momentum 0.9 without weight decay. In all experiments where achieving zero training error is possible, we continue training until the cross-entropy loss is less than 10^{-4}.

When calculating norms on a network with a Batch Normalization layer, we reparametrize the network to one that represents the exact same function without Batch Normalization as suggested in [19]. In all our figures we plot norm divided by margin to avoid scaling issues (see Section 2), where we set the margin over training set S to be 5^{th}-percentile of the margins of the data points in S, i.e. $\text{Prc}_5 \left\{ f_{\mathbf{w}}(x_i)[y_i] - \max_{y \neq y_i} f_{\mathbf{w}}(x)[y] | (x_i, y_i) \in S \right\}$. We have also investigated other versions of the margin and observed similar behavior to this notion.

We calculate the sharpness, as suggested in [11] - for each parameter w_i we bound the magnitude of perturbation by $\alpha(|w_i| + 1)$ for $\alpha = 5.10^{-4}$. In order to compute the maximum perturbation (maximize the loss), we perform 2000 updates of stochastic gradient ascent starting from the minimum, with mini-batch size 64, fixed step size 0.01 and momentum 0.9.

To compute the expected sharpness, we perturb each parameter w_i of the model with noise generated from Gaussian distribution with zero mean and standard deviation, $\alpha(10\,|w_i| + 1)$. The expected

sharpness is average over 1000 random perturbations each of which are averaged over a mini-batch of size 64. We compute the expected sharpness for different choices of α. For each value of α the KL divergence can be calculated as $\frac{1}{\alpha^2}\sum_i \left(\frac{w_i}{(10|w_i|+1)}\right)^2$.

B Proofs

B.1 Proof of Lemma 1

Proof. Define $g_{\{W_{-i-j},\boldsymbol{\nu}_{i,j}\}}(x)$ as the network $f_{\mathbf{w}}$ with weights in layers $i, j,, W_i, W_j$ replaced by $\boldsymbol{\nu}_i, \boldsymbol{\nu}_j$. Hence,

$$\begin{aligned}\|(W+\boldsymbol{\nu})_d\left(\Pi_{i=1}^{d-1}D_i(W+\boldsymbol{\nu})_i\right)*x - W_d\left(\Pi_{i=1}^{d-1}D_iW_i\right)*x\|_F \\ \leq \|\sum_i g(\{W_{-i},\boldsymbol{\nu}_i\},x)\|_F + \|\sum_{i,j} g(\{W_{-i-j},\boldsymbol{\nu}_{i,j}\},x)\|_F + \cdots + \|f_{\boldsymbol{\nu}}(x)\|_F \quad (8)\end{aligned}$$

Base case: First we show the bound for terms with one noisy layer. Let $g(\{W_{-k},\boldsymbol{\nu}_k\},x)$ denote $f_{\mathbf{w}}(x)$ with weights in layer k, W_k replaced by $\boldsymbol{\nu}_k$. Now notice that,

$$\begin{aligned}\mathbb{E}\|g(\{W_{-k},\boldsymbol{\nu}_k\},x)\|_F &= \mathbb{E}\|W_d\Pi_{i=k+1}^{d-1}D_iW_i * D_k\boldsymbol{\nu}_k * \left(\Pi_{i=1}^{k-1}D_iW_i\right)*x\|_F \\ &\overset{(i)}{\leq} \sigma_k\|W_d\Pi_{i=k+1}^{d-1}D_iW_i\|_F\|\left(\Pi_{i=1}^{k-1}D_iW_i\right)*x\|_F \\ &\overset{(ii)}{\leq} \sigma_k\frac{\sqrt{h_kh_{k-1}}}{\mu^2\|D_kW_k\|_F}\|W_d\left(\Pi_{i=1}^{d-1}D_iW_i\right)*x\|_F \\ &= \sigma_k\frac{\sqrt{h_kh_{k-1}}}{\mu^2\|D_kW_k\|_F}\|f_{\mathbf{w}}(x)\|_F.\end{aligned}$$

(i) follows from Lemma 3. (ii) follows from condition $C1$.

Induction step: Let for any set $s\subset[d], |s|=k$, the following holds:

$$\mathbb{E}\|g(\{W_{-i},\boldsymbol{\nu}_i\}_{i\in s},x)\|_F \leq \|f_{\mathbf{w}}(x)\|_F\Pi_{i\in s}\sigma_i\frac{\sqrt{h_ih_{i-1}}}{\mu^2\|D_iW_i\|_F}.$$

We will prove this now for terms with $k+1$ noisy layers.

$$\begin{aligned}\mathbb{E}\|g(\{W_{-i},\boldsymbol{\nu}_i\}_{i\in s\cup\{j\}},x)\|_F &\leq \sigma_j\frac{\sqrt{h_jh_{j-1}}}{\mu^2\|D_jW_j\|}\mathbb{E}\|g(\{W_{-i},\boldsymbol{\nu}_i\}_{i\in s},x)\|_F \\ &\leq \sigma_j\frac{\sqrt{h_jh_{j-1}}}{\mu^2\|D_jW_j\|}\|f_{\mathbf{w}}(x)\|_F\Pi_{i\in s}\sigma_i\frac{\sqrt{h_ih_{i-1}}}{\mu^2\|D_iW_i\|_F} \\ &= \|f_{\mathbf{w}}(x)\|_F\Pi_{i\in s\cup\{j\}}\sigma_i\frac{\sqrt{h_ih_{i-1}}}{\mu^2\|D_iW_i\|_F}\end{aligned}$$

Substituting the above expression in equation (8) gives,

$$\begin{aligned}\|(W+\boldsymbol{\nu})_d\left(\Pi_{i=1}^{d-1}D_i(W+\boldsymbol{\nu})_i\right)*x - W_d\left(\Pi_{i=1}^{d-1}D_iW_i\right)*x\|_F \\ \leq \left(\Pi_{i=1}^{d}\left(1+\frac{\sigma_i\sqrt{h_i}\sqrt{h_{i-1}}}{\mu^2\|D_iW_i\|_F}\right)-1\right)\|f_{\mathbf{w}}(x)\|_F.\end{aligned}$$

□

B.2 Proof of Lemma 2

Proof. We prove this lemma by induction on k. Recall that $\widehat{D}_i$ is the diagonal matrix with 0's and 1's corresponding to the activation pattern of the perturbed network $f_{\mathbf{w}+\boldsymbol{\nu}}(x)$. Let 1_E denote the

indicator function, that is 1 if the event E is true, 0 else. We also use $f^k_{\mathbf{w}}(x)$ to denote the network truncated to level k, in particular $f^k_{\mathbf{w}}(x) = \Pi_{i=1}^{k} D_k W_k x$.

Base case:

$$\|\widehat{D}_1 - D_1\|_1 = \sum_i 1_{\langle (W+\boldsymbol{\nu})_{1,i},x\rangle * \langle W_{1,i},x\rangle < 0} = \sum_i 1_{\langle (\mathbf{w})_{1,i},x\rangle^2 < -\langle (\boldsymbol{\nu})_{1,i},x\rangle * \langle (\mathbf{w})_{1,i},x\rangle} \leq \sum_i 1_{|\langle (\mathbf{w})_{1,i},x\rangle| < |\langle (\boldsymbol{\nu})_{1,i},x\rangle|}.$$

Since $\boldsymbol{\nu}_1$ is a random Gaussian matrix, and $\|x\| \leq 1$, for any i, $|\langle (\boldsymbol{\nu})_{1,i}, x\rangle| \leq \sigma_1(1+\delta_1)\sqrt{2\ln(h_1)}$ with probability greater than $1-\delta_1$. Hence, with probability $\geq 1-\delta_1$,

$$\|\widehat{D}_1 - D_1\|_1 \leq \sum_i 1_{|\langle (\mathbf{w})_{1,i},x\rangle| \leq \sigma_1\sqrt{20\ln(h_1)}} \leq C_2 h_1 \sigma_1 (1+\delta_1)\sqrt{2\ln(h_1)} = C_2 h_1 \sigma_1 C_{\delta_1}.$$

This completes the base case for $k=1$. $\widehat{D}_1$ is a random variable that depends on $\boldsymbol{\nu}_1$. Hence, in the remainder of the proof, to avoid this dependence, we separately bound $\widehat{D}_1 - D$ using the expression above and compute expectation only with respect to $\boldsymbol{\nu}_1$. With probability $\geq 1-\delta_1$,

$$\begin{aligned}
\mathbb{E}\|Err_1\|_F &= \mathbb{E}\|\widehat{D}_1 * (W+\boldsymbol{\nu})_1 x - D_1 * (W+\boldsymbol{\nu})_1 x\|_F \\
&\leq \mathbb{E}\|(\widehat{D}_1 - D_1) * W_1 x\|_F + \mathbb{E}\|(\widehat{D}_1 - D_1) * \boldsymbol{\nu}_1 x\|_F \\
&\overset{(i)}{\leq} \sqrt{C_2 h_1 \sigma_1 C_{\delta_1}}\sigma_1 + \sqrt{C_2 h_1 \sigma_1 C_{\delta_1}}\sigma_1 \\
&= 2\sqrt{C_2 h_1 \sigma_1 C_{\delta_1}}\sigma_1.
\end{aligned}$$

(i) follows because, each hidden node in $\mathbb{E}\|(\widehat{D}_1 - D_1) * W_1 x\|_F$ has norm less than $\sigma_1 C_{\delta_1}$ (as it changed its activation), number of such units is less than $C_2 h_1 \sigma_1 C_{\delta_1}$.

$k=1$ case does not capture all the intricacies and dependencies of higher layer networks. Hence we also evaluate the bounds for $k=2$.

$$\|\widehat{D}_2 - D_2\|_1 \leq \sum_i 1_{\langle (W+\boldsymbol{\nu})_{2,i}, f^1_{\mathbf{w}+\boldsymbol{\nu}}\rangle * \langle W_{2,i}, f^1_{\mathbf{w}}\rangle \leq 0} \leq \sum_i 1_{|\langle W_{2,i}, f^1_{\mathbf{w}}\rangle| \leq |\langle \boldsymbol{\nu}_{2,i}, f^1_{\mathbf{w}+\boldsymbol{\nu}}\rangle| + |\langle W_{2,i}, f^1_{\mathbf{w}+\boldsymbol{\nu}} - f^1_{\mathbf{w}}\rangle|}$$

Let $C_{\delta_2} = (1+\delta_2)\sqrt{2\ln(h_2)}$. Then, with probability $\geq 1-\delta_1-\delta_2$,

$$\begin{aligned}
&\left|\langle \boldsymbol{\nu}_{2,i}, f^1_{\mathbf{w}+\boldsymbol{\nu}}\rangle\right| + \left|\langle W_{2,i}, f^1_{\mathbf{w}+\boldsymbol{\nu}} - f^1_{\mathbf{w}}\rangle\right| \\
&\leq C_{\delta_2}\sigma_2\left(\|f^1_{\mathbf{w}}\|_F + 2\sqrt{C_2 h_1 \sigma_1 C_{\delta_1}}\sigma_1\right) + \|W_{2,i}\| 2\sqrt{C_2 h_1 \sigma_1 C_{\delta_1}}\sigma_1 \\
&\leq C_{\delta_2}\sigma_2\left(\|f^1_{\mathbf{w}}\|_F + 2\sqrt{C_2 h_1 \sigma_1 C_{\delta_1}}\sigma_1\right) + C_3\frac{\|D_2 W_2\|_F}{\sqrt{h_2}} 2\sqrt{C_2 h_1 \sigma_1 C_{\delta_1}}\sigma_1 \\
&\overset{(i)}{\leq} C_{\delta_2}\sigma_2\left(\|f^1_{\mathbf{w}}\|_F + 2\sqrt{\frac{\hat{\sigma_1}}{\sqrt{h_i + h_{i-1}}}}\hat{\sigma_1}\right) + 2\hat{\sigma_1}\frac{C_3\|f_{\mathbf{w}}(x)\|_F^{1/d}}{\mu}\sqrt{\frac{\hat{\sigma_1}}{\sqrt{h_i+h_{i-1}}}} \\
&= C_{\delta_2}\sigma_2\left(\|f^1_{\mathbf{w}}\|_F + \beta_1\hat{\sigma_1}\right) + \frac{C_3\|f_{\mathbf{w}}(x)\|_F^{1/d}}{\mu}\beta_1\hat{\sigma_1}
\end{aligned}$$

where, $\beta_i = 2\sqrt{\frac{\hat{\sigma_1}}{\sqrt{h_i+h_{i-1}}}}$. (i) follows from condition $C1$, which results in $\Pi_{i=2}^{d}\frac{\mu\|D_i W_i\|_F}{\sqrt{h_i}}\frac{\mu\|D_1 W_1 x\|_F}{\sqrt{h_1}} \leq \|f_{\mathbf{w}}(x)\|_F$. Hence, if we consider the rebalanced network[5] where all

[5]The parameters of ReLu networks can be scaled between layers without changing the function

layers have same values for $\frac{\mu\|D_iW_i\|_F}{\sqrt{h_i}}$, we get, $\frac{\mu\|D_iW_i\|_F}{\sqrt{h_i}} \leq \|f_{\mathbf{w}}(x)\|_F^{1/d}$. Also the above equations follow from setting, $\sigma_i = \frac{\hat{\sigma}_i}{C_2 C_{\delta_i}\sqrt{h_i+h_{i-1}}}$.

Hence, with probability $\geq 1-\delta_1-\delta_2$,

$$\|\widehat{D}_2 - D_2\|_1 \leq C_2 * h_2 \left(C_{\delta_2}\sigma_2 \left(\|f^1_{\mathbf{w}}\|_F + \beta_1\hat{\sigma_1}\right) + \frac{C_3\|f_{\mathbf{w}}(x)\|_F^{1/d}}{\mu}\beta_1\hat{\sigma_1} \right).$$

Since, we choose σ_i to scale as some small number $O(\sigma)$, in the above expression the first term scales as $O(\sigma)$ and the last two terms decay atleast as $O(\sigma^{3/2})$. Hence we do not include them in the computation of Err.

$$\begin{aligned}
\mathbb{E}\|Err_2\|_F &= \mathbb{E}\|\widehat{D}_2(W+\boldsymbol{\nu})_2 * \widehat{D}_1 * (W+\boldsymbol{\nu})_1 x - D_2(W+\boldsymbol{\nu})_2 * D_1 * (W+\boldsymbol{\nu})_1 x\|_F \\
&\leq \mathbb{E}\|(\widehat{D}_2 - D_2)(W+\boldsymbol{\nu})_2 * (\widehat{D}_1 - D_1) * (W+\boldsymbol{\nu})_1 x\|_F + \mathbb{E}\|D_2(W+\boldsymbol{\nu})_2 * (\widehat{D}_1 - D_1) * (W+\boldsymbol{\nu})_1 x\|_F \\
&\quad + \mathbb{E}\|(\widehat{D}_2 - D_2)(W+\boldsymbol{\nu})_2 * D_1 * (W+\boldsymbol{\nu})_1 x\|_F.
\end{aligned}$$

We will bound now the first term in the above expression. With probability $\geq 1-\delta_1-\delta_2$,

$$\begin{aligned}
&\mathbb{E}\|(\widehat{D}_2 - D_2)(W+\boldsymbol{\nu})_2 * (\widehat{D}_1 - D_1) * (W+\boldsymbol{\nu})_1 x\|_F \\
&\leq \mathbb{E}\|(\widehat{D}_2 - D_2)W_2 * (\widehat{D}_1 - D_1) * W_1 x\|_F + \mathbb{E}\|(\widehat{D}_2 - D_2)W_2 * (\widehat{D}_1 - D_1) * \boldsymbol{\nu}_1 x\|_F \\
&\quad + \mathbb{E}\|(\widehat{D}_2 - D_2)\boldsymbol{\nu}_2 * (\widehat{D}_1 - D_1) * W_1 x\|_F + \mathbb{E}\|(\widehat{D}_2 - D_2)\boldsymbol{\nu}_2 * (\widehat{D}_1 - D_1) * \boldsymbol{\nu}_1 x\|_F \\
&\leq 2\sqrt{C_2 * h_2 C_{\delta_2}\sigma_2\|f^1_W\|_F} C_{\delta_2}\sigma_2 \|f^1_W\|_F \sqrt{C_2 * h_1 * C_{\delta_1}\sigma_1} C_{\delta_1}\sigma_1 \\
&\quad + 2\sqrt{C_2 * h_2 C_{\delta_2}\sigma_2\|f^1_W\|_F} C_{\delta_2}\sigma_2\sqrt{h_1}\sqrt{C_2 * h_1 * C_{\delta_1}\sigma_1} C_{\delta_1}\sigma_1 + O(\sigma^2) \\
&\leq 4\|f^2_{\mathbf{w}}\|_F \frac{C_{\delta_2}\sigma_2 C_{\delta_1}\sigma_1\sqrt{h_1}}{\mu\|D_2W_2\|_F}\Pi_{i=1}^2\sqrt{C_2 h_i C_{\delta_i}\sigma_i}.
\end{aligned}$$

Induction step:

Now we assume the statement for all $i \leq k$ and prove it for $k+1$. $\|\widehat{D}_k - D_k\|_1 \leq O\left(C_2 h_k C_{\delta_k}\sigma_k \|f^{k-1}_{\mathbf{w}}\|_F\right)$ and $\mathbb{E}\|Err_k\|_F \leq O\left(\Pi_{i=1}^k \left(1 + \frac{\sigma_i\sqrt{h_i}\sqrt{h_{i-1}}C_2C_3}{\mu^2\|W_i\|_F}\right)\left(\Pi_{i=1}^k(1 + \frac{\sigma_i\sqrt{h_i}\sqrt{h_{i-1}}C_{\delta_i}C_2}{\mu^2\|W_i\|_F}) - 1\right)\|f^k_{\mathbf{w}}\|_F\right)$. Now we prove the statement for $k+1$.

$$\begin{aligned}
\|\widehat{D}_{k+1} - D_{k+1}\|_1 &= \sum_i 1_{\left\langle (W+\boldsymbol{\nu})_{k+1,i}, \Pi_{i=1}^k \widehat{D}_i(W+\boldsymbol{\nu})_i * x\right\rangle * \langle W_{2,i}, D_1W_1x\rangle \leq 0} \\
&\leq \sum_i 1_{|\left\langle W_{k+1,i}, \Pi_{i=1}^k \widehat{D}_i(W+\boldsymbol{\nu})_i * x\right\rangle| \leq |\left\langle \boldsymbol{\nu}_{k+1,i}, \Pi_{i=1}^k \widehat{D}_i(W+\boldsymbol{\nu})_i * x\right\rangle|} \\
&= \sum_i 1_{|\left\langle W_{k+1,i}, f^k_{\mathbf{w}+\boldsymbol{\nu}}\right\rangle| \leq |\left\langle \boldsymbol{\nu}_{k+1,i}, f^k_{\mathbf{w}+\boldsymbol{\nu}}\right\rangle|} \\
&\leq \sum_i 1_{|\left\langle W_{k+1,i}, f^k_W\right\rangle| \leq |\langle \boldsymbol{\nu}_{k+1,i}, f^k_{\mathbf{w}}\rangle| + |\left\langle \boldsymbol{\nu}_{k+1,i}, f^k_{\mathbf{w}+\boldsymbol{\nu}} - f^k_{\mathbf{w}}\right\rangle| + |\left\langle W_{k+1,i}, f^k_{\mathbf{w}+\boldsymbol{\nu}} - f^k_{\mathbf{w}}\right\rangle|}
\end{aligned}$$

Hence, with probability $\geq 1 - \sum_{i=1}^k \delta_i$,

$$\begin{aligned}
\|\widehat{D}_{k+1} - D_{k+1}\|_1 &\leq C_2 h_{k+1}\left[C_{\delta_k}\sigma_{k+1}(\|f^k_{\mathbf{w}}\|_F + \|f^k_{\mathbf{w}+\boldsymbol{\nu}} - f^k_{\mathbf{w}}\|_F) + \|W_{k+1,i}\|\|f^k_{\mathbf{w}+\boldsymbol{\nu}} - f^k_{\mathbf{w}}\|_F\right] \\
&\leq C_2 h_{k+1} C_{\delta_k}\sigma_{k+1}\|f^k_{\mathbf{w}}\|_F + C_2 h_{k+1} C_{\delta_k}\sigma_{k+1}\|f^k_{\mathbf{w}+\boldsymbol{\nu}} - f^k_{\mathbf{w}}\|_F + C_2 h_{k+1}\|W_{k+1,i}\|\|f^k_{\mathbf{w}+\boldsymbol{\nu}} - f^k_{\mathbf{w}}\|_F.
\end{aligned}$$

Now we will show that the last two terms in the above expression scale as $O(\sigma^2)$. For that, first notice that $\|f^k_{\mathbf{w}+\boldsymbol{\nu}} - f^k_{\mathbf{w}}\|_F \leq \left(\Pi_{i=1}^k \left(1 + \frac{\sigma_i\sqrt{h_i h_{i-1}}}{\mu^2\|D_i W_i\|_F}\right) - 1\right)\|f_{\mathbf{w}}(x)\|_F + Err_k$, from lemma 1. Note that the second term in the above expression clearly scale as $O(\sigma^2)$.

Hence,

$$\|\widehat{D}_{k+1} - D_{k+1}\|_1 \leq O\left(C_2 h_{k+1} C_{\delta_k}\sigma_{k+1}\|f^k_{\mathbf{w}}\|_F\right).$$

$$\begin{aligned}
\|Err_{k+1}\| &= \|f^{k+1}_{\mathbf{w}+\boldsymbol{\nu}} - \tilde{f}^{k+1}_{\mathbf{w}+\boldsymbol{\nu}}\|_F \\
&= \|\widehat{D}_{k+1}(W+\boldsymbol{\nu})_{k+1}\Pi_{i=1}^{k+1}\widehat{D}_i(W+\boldsymbol{\nu})_i x - D_{k+1}(W+\boldsymbol{\nu})_{k+1}\Pi_{i=1}^{k+1}D_i(W+\boldsymbol{\nu})_i x\|_F \\
&\leq \|(\widehat{D}_{k+1} - D_{k+1})(W+\boldsymbol{\nu})_{k+1}\Pi_{i=1}^{k+1}D_i(W+\boldsymbol{\nu})_i x\|_F + \|\widehat{D}_{k+1}(W+\boldsymbol{\nu})_{k+1}Err_k\|_F \\
&\leq \|(\widehat{D}_{k+1} - D_{k+1})(W+\boldsymbol{\nu})_{k+1}\Pi_{i=1}^{k+1}D_i(W+\boldsymbol{\nu})_i x\|_F + \|(\widehat{D}_{k+1} - D_{k+1})(W+\boldsymbol{\nu})_{k+1}Err_k\|_F \\
&\quad + \|D_{k+1}(W+\boldsymbol{\nu})_{k+1}Err_k\|_F
\end{aligned}$$

Substituting the bounds for $\widehat{D}_{k+1} - D_{k+1}$ and Err_k gives us, with probability $\geq 1 - \sum_{i=1}^{k+1}\delta_i$.

$$\begin{aligned}
\mathbb{E}\|Err_{k+1}\| \leq{}& \sqrt{C_2 h_{k+1} C_{\delta_k}\sigma_{k+1}\|f^k_{\mathbf{w}}\|_F} C_{\delta_k}\sigma_{k+1}\|f^k_W\|_F \mathbb{E}\|\Pi_{i=1}^{k+1}D_i(W+\boldsymbol{\nu})_i x\|_F \\
&+ \mathbb{E}\|Err_k\|_F\left(\sqrt{C_2 h_{k+1} C_{\delta_k}\sigma_{k+1}\|f^k_{\mathbf{w}}\|_F} C_{\delta_k}\sigma_{k+1}\|f^k_{\mathbf{w}}\|_F + \|D_{k+1}W_{k+1}\|_F + \sigma_{k+1}\sqrt{h_{k+1}}\right)
\end{aligned}$$

Now we bound the above terms following the same approach as in proof of Lemma 1, by considering all possible replacements of W_i with $\boldsymbol{\nu}_i$. That gives us the result. □

C Supporting results

Lemma 3. *Let A ,B be $n_1 \times n_2$ and $n_3 \times n_4$ matrices and $\boldsymbol{\nu}$ be a $n_2 \times n_3$ entrywise random Gaussian matrix with $\boldsymbol{\nu}_{ij} \sim \mathcal{N}(0, \sigma)$. Then,*

$$\mathbb{E}\left[\|A * \boldsymbol{\nu} * B\|_F\right] \leq \sigma\|A\|_F\|B\|_F.$$

Proof. By Jensen's inequality,

$$\begin{aligned}
\mathbb{E}\left[\|A * \boldsymbol{\nu} * B\|_F\right]^2 &\leq \mathbb{E}\left[\|A * \boldsymbol{\nu} * B\|_F^2\right] \\
&= \mathbb{E}\left[\left(\sum_{ij}\sum_{kl} A_{ik}\boldsymbol{\nu}_{kl}B_{lj}\right)^2\right] \\
&= \sum_{ij}\sum_{kl} A_{ik}^2\mathbb{E}\left[\boldsymbol{\nu}_{kl}^2\right]B_{lj}^2 \\
&= \sigma^2\|A\|_F^2\|B\|_F^2.
\end{aligned}$$

□

D Conditions in Theorem 1

In this section, we compare the conditions in Theorem 1 of a learned network with that of its random initialization. We trained a 10-layer feedforward network with 1000 hidden units in each layer on MNIST dataset. Figures 6, 7 and 8 compare condition $C1$, $C2$ and $C3$ on learned weights to that of random initialization respectively. Interestingly, we observe that the network with learned weights is very similar to its random initialization in terms of these conditions.

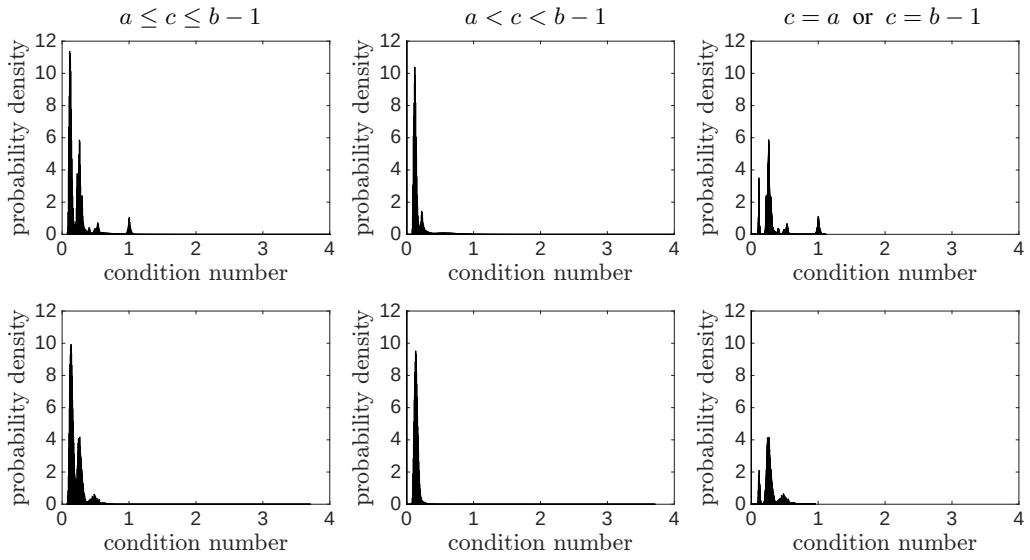

Figure 6: Condition $C1$: condition number $\frac{1}{\mu}$ of the network and its decomposition to two cases for random initialization and learned weights. **Top**: random initialization **Bottom**: learned weights. **Left**: distribution of all combinations of $a \leq c \leq b-1$. **Middle**: when $a < c < b-1$. **Right**: when $c = a$ or $c = b-1$.

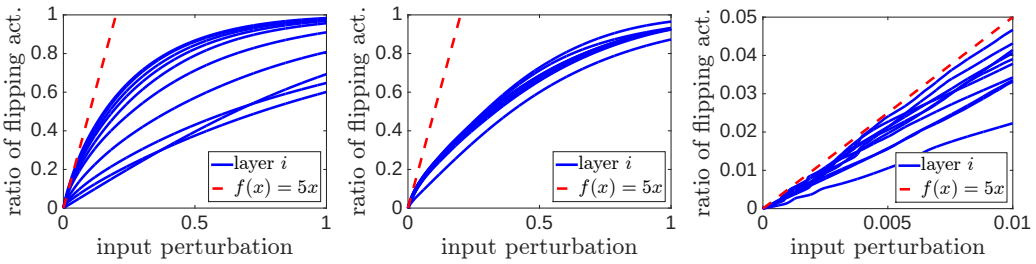

Figure 7: Ratio of activations that flip based on the magnitude of perturbation. **Left**: random initialization. **Middle**: learned weights. **Right**: learned weights (zoomed in).

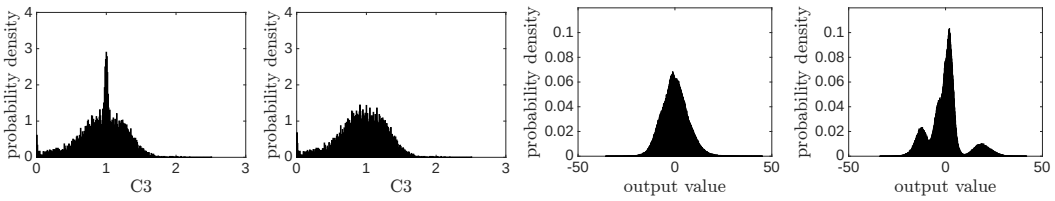

Figure 8: From left to right: Condition $C3$ for random initialization and learned network, output values for random and learned network