Heavy-Tailed Universality Predicts Trends in Test Accuracies for Very Large Pre-Trained Deep Neural Networks

Charles H. Martin* Michael W. Mahoney†

Abstract

Given two or more Deep Neural Networks (DNNs) with the same or similar architectures, and trained on the same dataset, but trained with different solvers, parameters, hyper-parameters, regularization, etc., can we predict which DNN will have the best test accuracy, and can we do so without peeking at the test data? In this paper, we show how to use a new Theory of Heavy-Tailed Self-Regularization (HT-SR) to answer this. HT-SR suggests, among other things, that modern DNNs exhibit what we call Heavy-Tailed Mechanistic Universality (HT-MU), meaning that the correlations in the layer weight matrices can be fit to a power law (PL) with exponents that lie in common Universality classes from Heavy-Tailed Random Matrix Theory (HT-RMT). From this, we develop a Universal capacity control metric that is a weighted average of PL exponents. Rather than considering small toy NNs, we examine over 50 different, large-scale pre-trained DNNs, ranging over 15 different architectures, trained on ImagetNet, each of which has been reported to have different test accuracies. We show that this new capacity metric correlates very well with the reported test accuracies of these DNNs, looking across each architecture (VGG16/.../VGG19, ResNet10/.../ResNet152, etc.). We also show how to approximate the metric by the more familiar Product Norm capacity measure, as the average of the log Frobenius norm of the layer weight matrices. Our approach requires no changes to the underlying DNN or its loss function, it does not require us to train a model (although it could be used to monitor training), and it does not even require access to the ImageNet data.

1 Introduction

We are interested in the following general question.

- Given two or more Deep Neural Networks (DNNs) with the same or similar architectures, trained on the same dataset, but trained with different solvers, parameters, hyper-parameters, regularization, etc., can we predict which DNN will have the best test accuracy, and can we do so without peeking at the test data?

This question is both theoretical and practical. Theoretically, solving this would help to understand why this class of machine learning (ML) models performs as well as it does in certain classes of applications. Practically, there are many motivating examples. Here are two.

- **Automating architecture search.** Developing DNN models requires significant architecture engineering, so there is interest in automating the design of DNNs. Current methods can produce a series of DNNs subject to given general architecture constraints, but the models must be evaluated using cross validation (CV). DNNs have so many adjustable parameters that even when using CV it is possible to leak information from the test sets into the training data, thus producing brittle, non-robust models. It is thus of interest to have design principles and quality metrics that do not depend on the test data and/or the labels.

- **Fine-Tuning Pre-trained Models.** Since one often does not have enough labeled data to train a large DNN from scratch, many modern engineering solutions can re-use widely-available pre-trained DNNs, fine-tuning them on smaller data sets. This technique often works extremely well for visual tasks, using DNNs pre-trained on ImageNet; and recently it has become feasible for complex natural language processing (NLP) tasks. Sometimes, however, these fine-tuned models become brittle and non-robust—due to overtraining, because information leaks from the test set into the training data. Here, it would also be very helpful to be able to fine-tune large, pre-trained DNNs without needing to peek at the test data.

To predict trends in the generalization accuracy of a series of DNN architectures, VC-like theories offer theoretical bounds on the generalization accuracy. Practically, such capacity metrics can guide the theoretical development of new regularizers for traditional ML optimization problems (e.g., counterfactual expected risk minimization [1]), but the bounds themselves are far too loose to be used directly. Moreover, since the

*Calculation Consulting, 8 Locksley Ave, 6B, San Francisco, CA 94122. charles@CalculationConsulting.com

†ICSI and Department of Statistics, University of California at Berkeley, Berkeley, CA 94720, mmahoney@stat.berkeley.edu

early days of NN research, it was known that VC theory could (probably) not be directly applied to the seemingly wildly non-convex optimization problem implicitly posed by NNs. (This has caused some researchers to suggest we need to rethink regularization in DNNs entirely.)

In light of this, Liao et al. [2] used an appropriately-scaled, data-dependent Product Norm capacity control metric to bound the worst-case generalization error for several small (non production-quality, but still interesting) DNN models, and they showed that the bounds are remarkably tight. There is, in fact, a large body of work on norm-based capacity control metrics, both recent, e.g., [2, 3, 4] and [5, 6, 7, 8, 9, 10, 11, 12, 13, 14], as well as much older [15, 16]. Much of this work has been motivated by the observation that parameter counting and more traditional VC-based bounds tend to lead to vacuous results for modern state-of-the-art DNNs, e.g., since modern DNNs are heavily over-parameterized and depend so strongly on the training data.

As with most theoretical studies, Liao et al.'s approach and intent differ greatly from ours. They seek *worst-case* complexity bounds, motivated to reconcile discrepancies with more traditional statistical learning theory, and they apply them (to quite small-scale NNs). To address our main question, we seek an *average-case* or *typical case* (for realistic large-scale NNs) complexity metric, viable in production to guide the development of better DNNs at scale. Bounding a small toy model does not necessarily mean that the individual weight matrix norms in production-quality DNNs will be directly comparable. In particular, it does not mean that we can directly compare the individual weight matrix norms across layers in different, and more complex, architectures. Also, Liao et al. had to modify the DNN optimization loss function. This means that their approach cannot be tested/evaluated on any existing *pre-trained* DNN architecture, e.g., the VGG and ResNet models, widely-used today in industry. Still, their results do *suggest* that a Product Norm may work well as a *practical* capacity metric for large, and perhaps even pre-trained, production-quality DNNs. We will evaluate this and show that it does. More generally, to predict trends in the test accuracies, one needs some more *Universal* empirical metric that transfers across DNN architectures.

Recent work by Martin and Mahoney [17, 18] suggests a Universal empirical metric to characterize the amount of *Implicit Self-Regularization* and, accordingly, the generalization capacity, for a wide range of pre-trained DNNs.[1] The metric (defined below) involves the power law (PL) exponents, α, of individual layer weight matrices, $\mathbf{W}$, as determined by fitting the Empirical Spectral Density (ESD), $\rho(\lambda)$, to a PL distribution. Looking in detail at a series of models, like AlexNet, VGG, ResNet, etc, they observe that the (linear) layer weight matrices almost always follow a PL distribution, and fitted PL exponents nearly all lie within a universal range $\alpha \in [2, 5]$. Analysis of a small model (MinAlexNet) demonstrates that smaller PL exponents α correspond to better generalization. Subsequent work [19] demonstrated Heavy-Tailed (HT) behavior in nearly every pre-trained architecture studied, e.g., across nearly 7500 layer weight matrices (and 2D feature maps), including DNNs pre-trained for computer vision tasks on ImageNet, and for several different NLP tasks.

When one observes good empirical PL fits of ESDs of correlations of layer weight matrices, we say the DNN *exhibits Heavy-Tailed behavior*. Motivated by these empirical observations, and using the Universality properties of Heavy-Tailed Random Matrix Theory (HT-RMT), Martin and Mahoney developed a theory of Heavy-Tailed Self-Regularization (HT-SR) for DNNs [20, 17, 18]. We build on and extend that theory here.

In Statistical Physics, Universality of PL exponents is very non-trivial, and it suggests the presence of a deeper, underlying, *Universal mechanism* driving the system dynamics [21, 22]. It is this *Heavy Tailed Mechanistic Universality* (HT-MU), as we call it, that originally motivated our study. HT-MU applies to the analysis of complicated systems, including many physical systems, traditional NNs [23, 24], and even models of the dynamics of actual spiking neurons. Indeed, the dynamics of learning in DNNs seems to resemble a system near a phase transition, such as the phase boundary of spin glass, or a system displaying Self Organized Criticality (SOC), or a Jamming transition [25, 26]. Of course, we can not say which mechanism, if any, is at play. Instead, we use the machinery of HT-RMT as a stand-in for a generative model of the weight matrices in DNNs, to catalog and model the HT behavior of DNNs.[2] This Universality *suggests* that we look for a *Universal Capacity Control Metric*[3] to address our main question.

[1] A short version of [17] is available as [18]. The long version contains many more results and a much more detailed exposition.

[2] Perhaps the most well-known Universality in RMT is associated with the Gaussian Universality class, where the sum of many random variables drawn from a wide range of distributions is "approximately Gaussian," e.g., in the sense that the sum approaches a suitably-normalized Gaussian distribution. As briefly reviewed in Appendix A of [27], HT Universality makes analogous (but, admittedly, more complicated) statements for random variables drawn from distributions in which the tails decay more slowly than those in the Gaussian Universality class [17, 18].

[3] To be clear, this metric is Universal, not in the sense that it applies "universally" to every possible DNN, but in the Statistical Physics sense [21, 22] that it applies to matrices within/across HT "Universality" classes.

Our main results are the following.

- We evaluate the Product Norm capacity control metric on a wide range of large-scale pre-trained production-level DNNs, including VGG and ResNet series, demonstrating that it correlates well with reported average test accuracies across many series of models. *While norm-based metrics have been applied to small models, to our knowledge, evaluating this metric to predict trends in test accuracies of large-scale pre-trained models has never (until now) been reported.*

- We introduce a new methodology to analyze the performance of large-scale pre-trained DNNs, using a phenomena observed in HT-SR Theory. We construct a Universal capacity control metric to predict average DNN test performance. This metric is a weighted average of layer PL exponents, $\hat{\alpha}$, weighted by the $\log$[4] of the Spectral norm (i.e., maximum eigenvalue λ^{max}) of layer correlation matrices:

$$\hat{\alpha} = \sum_{l \in L} \alpha_l \log \lambda_l^{max}.$$

- We apply our Universal capacity control metric $\hat{\alpha}$ to a wide range of large-scale pre-trained production-level DNNs, including the VGG and ResNet series of models, as well as many others. This metric correlates very well with the reported average test accuracies across many series of pre-trained DNNs.

- We provide a derivation for a relation between our Universal capacity control metric $\hat{\alpha}$ and the well known Product Norm capacity control metric, i.e, in the form of the average log of the squared Frobenius norm:

$$\langle \log \|\mathbf{W}\|_F^2 \rangle \approx \frac{1}{N_L} \sum_{l \in L} \alpha_l \log \lambda_l^{max}.$$

  We do not make precise the error in "$\approx$" but our derivation makes clear that we expect the approximation to be good for smaller α and less good for larger α.

There is a tradeoff here: our $\hat{\alpha}$ metric has two parameters (α and λ^{max}), as opposed to the Product Norm capacity control metric, which has one ($\|\cdot\|_F^2$), and it is more expensive to compute, but it does perform better. Informally, as opposed to looking only at the "size" or "shape" of a model, e.g., as with a norm-based metric, the parameters λ_l^{max} and α_l take into consideration both the size and the shape of the model.

For both our Universal $\hat{\alpha}$ metric and the Product Norm metric, our empirical results are, to our knowledge, the first time such theoretical capacity metrics have been reported to predict (trends in) the test accuracy for *pre-trained production-level* DNNs. In particular, this illustrates the usefulness of these norm-based metrics beyond smaller models such as MNIST, CIFAR10, and CIFAR100. Our results, including for both our Universal metric and the Product Norm metric we consider, can be reproduced with the `WeightWatcher` package[5]; and our results suggest that our "practical theory" approach is fruitful more generally for engineering good algorithms for realistic large-scale DNNs.

A longer technical report version of this paper, with appendices to which we will refer, has appeared as [27].

2 Brief Overview of Heavy-Tailed Self-Regularization

Here, we briefly review Martin and Mahoney's Theory of Heavy-Tailed Self-Regularization (HT-SR) [17, 18]. See Appendix A of [27] for more details.

Write the Energy Landscape (or optimization function, parameterized by $\mathbf{W}_l$s and $\mathbf{b}_l$s) for a typical DNN with L layers, with activation functions $h_l(\cdot)$, and with $N \times M$ weight matrices $\mathbf{W}_l$ and biases $\mathbf{b}_l$, as:

$$E_{DNN} = h_L(\mathbf{W}_L \cdot h_{L-1}(\mathbf{W}_{L-1} \cdot h_{L-2}(\cdots) + \mathbf{b}_{L-1}) + \mathbf{b}_L).$$

Typically, this model would be trained on some labeled data $\{d_i, y_i\} \in \mathcal{D}$, using Backprop, by minimizing the loss $\mathcal{L}$. For simplicity, we do not indicate the structural details of the layers (e.g., Dense or not, Convolutions or not, Residual/Skip Connections, etc.).

In the HT-SR Theory, we analyze the eigenvalue spectrum (the ESD) of the associated correlation matrices [17, 18]. From this, we can characterize the amount and form of correlation (and therefore the implicit self-regularizartion) present in the DNN's weight matrices. For each layer matrix $\mathbf{W}_l$, of size $N \times M$, construct the associated $M \times M$ (uncentered) correlation matrix $\mathbf{X}_l$. Dropping the L and l, i indices, we have $\mathbf{X} = \frac{1}{N}\mathbf{W}^T\mathbf{W}$. If we compute the eigenvalue spectrum of $\mathbf{X}$, i.e., λ_i such that $\mathbf{X}\mathbf{v}_i = \lambda_i \mathbf{v}_i$, then the ESD of eigenvalues, $\rho(\lambda)$, is just a histogram of the eigenvalues. Using HT-SR Theory [17, 18], we can characterize the *correlations* in a weight matrix by examining its ESD, $\rho(\lambda)$. It can be well-fit to a power law (PL) distribution, given as $\rho(\lambda) \sim \lambda^{-\alpha}$, which is (at least) valid within a bounded range of eigenvalues $\lambda \in [\lambda^{min}, \lambda^{max}]$.

When we observe HT behavior in $\mathbf{W}$, or rather its correlation matrix $\mathbf{X}$, we essentially use HT-RMT as a generative model. We say that we *model* $\mathbf{W}$ *as if* it is a random matrix, $\mathbf{W}^{rand}(\mu)$, drawn from a Universality class of HT-RMT (i.e., VHT, MHT, or WHT, as defined below). To characterize this HT-MU behavior, we use

[4] Throughout, we use log base 10.

[5] `https://pypi.org/project/WeightWatcher/`

a HT variant of RMT and use HT random matrices to elucidate different Universality classes. Let $\mathbf{W}(\mu)$ be an $N \times M$ random matrix with entries chosen i.i.d. from

$$\mathbf{Pr}\left[W_{i,j}\right] \sim \frac{W_0^{\mu}}{|W_{i,j}|^{1+\mu}},$$

where W_0 is the typical order of magnitude of $W_{i,j}$, and where $\mu > 0$. There are at least 3 different Universality classes of HT random matrices, defined by the range μ takes on:

- $0 < \mu < 2$: VHT: Universality class of Very Heavy-Tailed (or Lévy) matrices;
- $2 < \mu < 4$: MHT: Universality class of Moderately Heavy-Tailed (or Fat-Tailed) matrices;
- $4 < \mu$: WHT: Universality class of Weakly Heavy-Tailed matrices.

3 Heavy-Tailed Mechanistic Universality and Capacity Control Metrics

From prior work [17, 18], we expect that smaller PL exponents of the ESD imply more regularization and therefore better generalization. Since smaller norms of weight matrices often correspond to better capacity control [2, 3, 4, 8], we would like to relate the empirical PL exponent α to the empirical Frobenius norm $\|\mathbf{W}\|_F$. At least naïvely, this is a challenge, since smaller PL exponents often correspond to larger matrix norms (and thus worse generalization!). See Appendices C and D of [27] for more details. To resolve this apparent discrepancy, we will exploit HT-MU to propose a Universal DNN complexity metric.

Form of a Proposed Universal DNN Complexity Metric. The PL exponent α is a complexity metric for a single DNN weight matrix, with smaller values corresponding to greater regularization [17, 18]. It describes how well that matrix encodes complex correlations in the training data. Thus, a natural class of complexity or capacity metrics to consider for a DNN is to take a *weighted average*[6] of the PL exponents, $\alpha_{l,i}$, for each layer weight matrix $\mathbf{W}_{l,i}$:

$$\hat{\alpha} := \frac{1}{N_L}\sum_{l,i} b_{l,i}\alpha_{l,i}. \tag{3.1}$$

[6]There are several reasons we don't want an unweighted average: an unweighted average behaves differently for HT random matrices than for well-trained DNN weight matrices, and so it would not be Universal; we want a metric that relates the α of HT-SR Theory with known capacity control metrics such as norms of weight matrices, and including weights permits this flexibility; we want weights to encode information that "larger" matrices are somehow more important; and unweighted averages, while sometimes providing predictive quality, do not perform as reliably well. See Appendices C and D of [27] for more details.

Here, the smaller $\hat{\alpha}$, the better we expect the DNN to represent training data, and (presumably) the better the DNN will generalize. The main question is: what are good weights $b_{l,i}$?

As we now show, we can extract the weighted average $\hat{\alpha}$ directly from the more familiar Product Norm, by exploiting both HT Universality, and its finite-size effects, arising in DNN weight matrices.

Product Norm Measures of Complexity. It has been suggested that the complexity, $\mathcal{C}$, of a DNN can be characterized by the product of the norms of layer weight matrices,

$$\mathcal{C} \sim \|\mathbf{W}_1\| \times \|\mathbf{W}_2\| \cdots \|\mathbf{W}_L\|,$$

where $\|\mathbf{W}\|$ is, e.g., the Frobenius norm [2, 3, 4]. (Here, we can use either $\|\mathbf{W}\|$ or $\|\mathbf{W}\|^2$, and one can view $\mathcal{C}$ as akin to a data-dependent VC complexity.) To that end, we consider a log complexity

$$\begin{aligned}\log \mathcal{C} &\sim \log\left[\|\mathbf{W}_1\| \times \|\mathbf{W}_2\| \cdots \|\mathbf{W}_L\|\right] \\ &\sim \left[\log\|\mathbf{W}_1\| + \log\|\mathbf{W}_2\| \cdots \log\|\mathbf{W}_L\|\right],\end{aligned}$$

and we define the average log norm of weight matrices (where N_L is the number of layers) as

$$\langle \log \|\mathbf{W}\| \rangle = \frac{1}{N_L}\sum_{l} \log \|\mathbf{W}_l\|. \tag{3.2}$$

A Universal, Linear, PL–Norm Relation. Based on our empirical results and theoretical considerations, we propose a simple linear relation between the (squared) Frobenius norm $\|\mathbf{W}\|_F^2$ of $\mathbf{W}$, the PL exponent α, and the maximum eigenvalue λ^{max} of $\mathbf{X}$ (i.e., the spectral norm $\|\mathbf{X}\|_2 = \frac{1}{N}\|\mathbf{W}\|_2^2$):

$$\textbf{PL–Norm Relation: } \alpha \log \lambda^{max} \approx \log \|\mathbf{W}\|_F^2. \tag{3.3}$$

To our knowledge, this is the first time this PL–Norm relation has been noted in the literature (although prior work has considered norm bounds for HT data [16]). A few comments on Eqn. (3.3). First, it provides a connection between the PL parameter α of HT-SR Theory and the weight norm $\|\mathbf{W}\|_F^2$ of more traditional statistical learning theory. Second, it has a structural form like that of the well-known Hausdorff dimension [28]. Third, it shows that PL exponents can alternatively be interpreted (up to the $\frac{1}{N}$ scaling) as the Stable Rank in Log-Units:

$$\text{Log-Units Stable Rank:} \quad \mathcal{R}_s^{log} := \frac{\log \|\mathbf{W}\|_F^2}{\log \lambda^{max}} \approx \alpha.$$

Our justification for proposing Eqn. (3.3) is three-fold.

1. We derive Eqn. (3.3) in the special case of very small PL exponent, $\alpha \to 1$ ($\mu \to 0$), for an $N \times M$ matrix $\mathbf{W}^{rand}(\mu)$ (with $N = M$, or $Q = 1$, where $Q = N/M$).[7]

2. For finite-size random matrices $\mathbf{W}^{rand}(\mu)$, we expect the MHT Universality class, $\mu \in (2, 4)$, to behave *like* the VHT Universality class, $\mu \in (1, 2)$. Because of this similarity, we expect that we can extend Eqn. (3.3), approximately, to larger PL exponents. For $N \sim \mathcal{O}(100 - 1000)$, $\alpha \log \lambda^{max}$ increases nearly linearly with $\log \|\mathbf{W}^{rand}(\mu)\|_F^2$ as μ increases. For larger N, the relation saturates for large μ. See Appendix B of [27].

3. *As evidence of HT-MU*, we observe empirically that Eqn. (3.3) also applies, approximately, to the real DNN weight matrices $\mathbf{W}$. We see that $\alpha \log \lambda^{max}$ is positively correlated with $\log \|\mathbf{W}\|_F^2$ as α increases, and even shows similar saturation effects at large α. See Appendix C of [27].

Finally, based on Eqn. (3.3), we choose the weights in Eqn. (3.1) to be the log of the corresponding maximum eigenvalues of $\mathbf{X}$. That is, for a given l, i, we have the weights in Eqn. (3.1) as

$$b_{l,i} = \lambda_{l,i}^{max}.$$

Then, we define the complexity metrics for Linear and Convolutional Layers as follows:

$$\begin{aligned} &\text{Linear Layer:} && \log \|\mathbf{W}_l\|_F^2 \to \alpha_l \log \lambda_l^{max} \\ &\text{Conv2D Layer:} && \log \|\mathbf{W}_l\|_F^2 \to \sum_{i=1}^{n_l} \alpha_{l,i} \log \lambda_{l,i}^{max}, \end{aligned}$$

where, for Conv2D Layers, we relate the "norm" of the 4-index Tensor $\mathbf{W}_l$ to the sum of the $n_l = c \times d$ terms for each feature map. This lets us compare the Product Norm to the weighted average of PL exponents as follows:

$$2 \log \mathcal{C} = \langle \log \|\mathbf{W}\|_F^2 \rangle \to \hat{\alpha} := \frac{1}{N_L} \sum_{i,l} \alpha_{i,l} \log \lambda_{l,i}^{max}. \tag{3.4}$$

Given these connections, in Section 4, we will use $\hat{\alpha}$ to analyze numerous pre-trained DNNs.

The PL–Norm Relation: Deriving a Special Case of Eqn. (3.3). Here, we derive Eqn. (3.3) in the special case of very small PL exponent, as $\mu \to 0$, for an $N \times M$ random matrix $\mathbf{W}$, with $M = N, Q = 1$, and with elements drawn from Eqn. (A.3) of [27].[8] We seek a relation good in the region $\mu \in [0, 2]$, and we will extend the $\mu \sim 0$ results to this full region. That is, we establish this as an asymptotic relation for the VHT Universality class for very small exponents.

To start, recall that $\|\mathbf{W}\|_F^2 = \text{Trace}[\mathbf{W}^T\mathbf{W}] = N \text{ Trace}[\mathbf{X}]$. Since, $\mu \gtrsim 0$, the eigenvalue spectrum is dominated by a single large eigenvalue, it follows that

$$\|\mathbf{W}\|_F^2 \approx N\lambda^{max},$$

where λ^{max} is the largest eigenvalue of the matrix $\mathbf{X}$ (with the $1/N$ normalization). Taking the log of both sides of this expression and expanding leads to

$$\log \|\mathbf{W}\|_F^2 \approx \log (N\lambda^{max}) = \log N + \log \lambda^{max}.$$

Rearranging, we get that

$$\frac{\log \|\mathbf{W}\|_F^2}{\log \lambda^{max}} \approx \frac{\log N}{\log \lambda^{max}} + 1.$$

Thus, for a parameter α satisfying Eqn. (3.3), we have

$$\alpha \approx \frac{\log N}{\log \lambda^{max}} + 1.$$

The relation between α and μ for the VHT Universality class is given in Eqn. (A.4a) of [27] as $\alpha = \frac{1}{2}\mu + 1$. Thus, to establish our result, we need to show that

$$\frac{\log N}{\log \lambda^{max}} \approx \frac{1}{2}\mu.$$

To do this, we use the relation of Eqn. (A.5) of [27] for the tail statistic, i.e., that $\lambda^{max} \approx N^{4/\mu - 1}$. Taking the log of both sides gives

$$\log \lambda^{max} \approx \log N^{4/\mu - 1} = (4/\mu - 1) \log N,$$

from which it follows that

$$\frac{\log N}{\log \lambda^{max}} \approx \frac{\log N}{(4/\mu - 1) \log N} = \frac{1}{4/\mu - 1}.$$

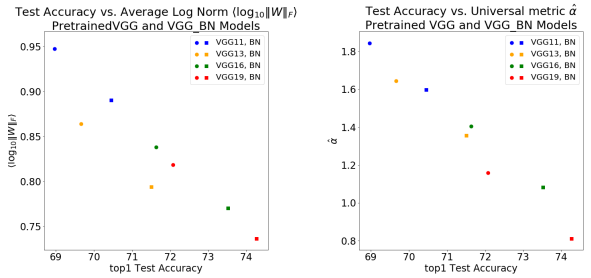
Finally, we can form the Taylor Series for $\dfrac{1}{4/\mu - 1}$ around, e.g., $\mu = 1.15 \approx 1$, which gives

$$\left.\frac{1}{4/\mu - 1}\right|_{\mu=1.15} \approx \frac{1}{2}\mu - \frac{1}{6} + \cdots \approx \frac{1}{2}\mu.$$

This establishes the approximate—and rather surprising—linear relation we want for $\mu \in [0, 2]$ for the VHT Universality class of HT-RMT.

[7] In particular, while this is a limiting statement, we expect to observe small deviations from this when we are not in the limit.

[8] We derive Eqn. (3.3) at what is sometimes pejoratively known as "at a physics level of rigor." That is fine, as our justification ultimately lies in our empirical results. Recall our goal: to derive a very simple expression relating fitted PL exponents and Frobenius norms that is usable by practical engineers working with state-of-the-art models, i.e., not simply small toy models. There is very little "rigorous" work on HT-RMT, less still on understanding finite-sized effects of HT Universality. Hopefully, our results will lead to more work along these lines.



(a) log Frobenius norm $\langle\log\|\mathbf{W}\|_F\rangle$ (b) weighted average PL exponent $\hat{\alpha}$

Figure 1: Pre-trained VGG and VGG_BN Architectures and DNNs. Top 1 Test Accuracy versus average log Frobenius norm $\langle\log\|\mathbf{W}\|_F\rangle$ (in (1(a))) or Universal, weighted average PL exponent $\hat{\alpha}$ (in (1(b))) for VGG11 vs VGG11_BN (blue), VGG13 vs VGG13_BN (orange), VGG16 vs VGG16_BN (green), and VGG19 vs VGG19_BN (red). We plot plain the VGG models with circles and the VGG_BN models with squares.

Model	Top1 Accuracy	$\hat{\alpha}$
VGG11	68.97	1.84
VGG11_BN	70.45	1.60
VGG13	69.66	1.65
VGG13_BN	71.51	1.36
VGG16	71.64	1.41
VGG16_BN	73.52	1.08
VGG19	72.08	1.16
VGG19_BN	74.27	0.81

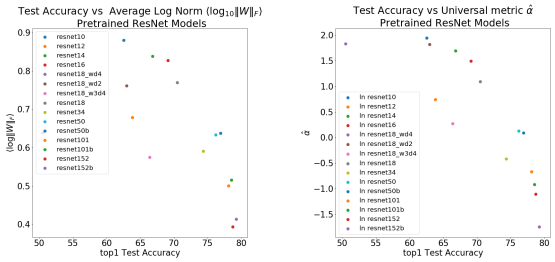
Table 1: Results for VGG Architecture. Top1 Accuracy is defined as the 100.0 minus the Top1 reported error.

4 Empirical Results on Pre-trained DNNs

Here, we summarize our empirical results. We only consider Linear and Conv2D layers because we only examine series of commonly available, open source, pre-trained DNNs with these kinds of layers. All models have been trained on ImageNet, and reported test accuracies are widely available. Throughout, we use Test Accuracies for the Top1 errors (where Accuracy = 100 - Top1 error). We see similar results for Top5 errors. We emphasize that, *for our analysis, we do not need to retrain these models—and we do not even need the test data!*

VGG and VGG_BN Models. We first look at the VGG class of models, comparing the log norm and the Universal $\hat{\alpha}$ metrics. See Figure 1 and Table 1 for a summary of the results. Figures 1(a) and 1(b) show both the average log Frobenius norm, $\langle\log\|\mathbf{W}\|_F\rangle$ of Eqn. (3.2), and the weighted average PL exponent, $\hat{\alpha}$ of Eqn. (3.4), as a function of the reported (Top1) test accuracy for the series of pre-trained VGG models, as available in the pyTorch package.[9] These models include VGG11, VGG13, VGG16, and VGG19, as well as their more accurate counterparts with Batch Normalization, VGG11_BN, VGG13_BN, VGG16_BN and VGG19_BN. Table 1 provides additional details.

Across the entire series of architectures, reported test accuracies increase linearly as each metric, $\langle\log\|\mathbf{W}\|_F\rangle$ and $\hat{\alpha}$, decreases. Moreover, whereas the log norm relation has 2 outliers, VGG13 and VGG13_BN, the Universal $\hat{\alpha}$ metric shows a near perfect linear relation across the entire VGG series.



(a) log Frobenius norm $\langle\log\|\mathbf{W}\|_F\rangle$ (b) weighted average PL exponent $\hat{\alpha}$

Figure 2: Pre-trained ResNet Architectures and DNNs. Top 1 Test Accuracy versus average log Frobenius norm $\langle\log\|\mathbf{W}\|_F\rangle$ (in (2(a))) or Universal, weighted average PL exponent $\hat{\alpha}$ (in (2(b))).

ResNet Models. We next look at the ResNet class of models. See Figure 2 and Table 2 for a summary of the results. Here, we consider a set of 15 different pre-trained ResNet models, of varying sizes and accuracies, ranging from the small ResNet10 up to the largest ResNet152 models, as provided by the OSMR sandbox,[10] developed for training large-scale image classification networks for embedded systems. Again, we compare the reported (Top1) test accuracy versus the average log norm $\langle\log\|\mathbf{W}\|_F\rangle$ and the Universal $\hat{\alpha}$ metrics.

As with the VGG series, both metrics monotonically decrease as test accuracies decrease for ResNet series, and both metrics have a few large outliers off the main line relation. See Figures 2(a) and 2(b). In particular, the log norm metric has several notable outliers, including resnet18_wd2, resnet18_wd3_d4, resnet34, and resnet10. The $\hat{\alpha}$ metric shows a slightly better relation, with resnet18_wd2 more in line, and the other 3 outliers a little less off the main line of correlation. The $\hat{\alpha}$ metric is as good or slightly better than average log norm metric for the Resnet series of models.

[9] https://pytorch.org/
[10] https://github.com/osmr/imgclsmob

Architecture	Model	Top1 Accuracy	$\hat{\alpha}$
ResNet (small)	resnet10	62.54	1.94
	resnet12	63.82	0.74
	resnet14	66.83	1.70
	resnet16	69.10	1.49
ResNet18	resnet18_wd4	50.50	1.83
	resnet18_wd2	62.96	1.82
	resnet18_w3d4	66.39	0.28
	resnet18	70.48	1.09
ResNet34	resnet34	74.34	-0.42
ResNet50	resnet50	76.21	0.13
	resnet50b	76.95	0.09
ResNet101	resnet101	78.10	-0.67
	resnet101b	78.55	-0.92
ResNet152	resnet152	78.74	-1.11
	resnet152b	79.26	-1.74

Table 2: Results for ResNet Architectures and DNN Models. The Top1 Accuracy is defined as the 100.0 minus the Top1 reported error. Some $\hat{\alpha} < 0$ because the of how the ResNet weight matrices are internally scale and normalized, which makes the maximum eigenvalue less then one, $\lambda^{max} < 1$.

We see similar results for our Universal PL capacity control metric $\hat{\alpha}$ across a wide range of other pre-trained DNN models, described in next. In nearly all cases, the metric $\hat{\alpha}$ correlates well with the reported test accuracies, with only a three DNN architectures as exceptions. Overall the $\hat{\alpha}$ metric systematically correlates well with the generalization accuracy of a wide class of pre-trained DNN architectures—which is rather remarkable.

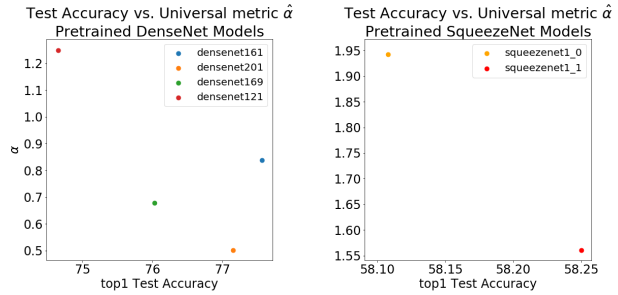
More Pre-trained Models. We present results for eleven more series of pre-trained DNN architectures, eight of which show positive results, as with the VGG and ResNet series, and three of which provide counterexample architectures. See Table 3 for a summary.

The results that perform as expected are show in Figures 3, 4, 5, and 6. For each set of models, our Universal metric $\hat{\alpha}$ is smaller when, for the most part, the reported (Top 1) test accuracy is larger. This holds approximately true for the three of the four DenseNet models, with densenet169 as an outlier. In fact, this is the only outlier out of 26 DNN models in these 8 architectures. For all of the other pre-trained DNNs, smaller $\hat{\alpha}$ corresponds with smaller test error and larger test accuracy, as predicted by our theory.

Counterexamples. In such a large corpus of DNNs, there are of course exceptions for a predictive theory. See Table 3 for the counterexamples. These are ResNeXt, MeNet, and FDMobileNet. For ResNeXt, there are only two models, and the $\hat{\alpha}$ is larger for the less accurate model. For MeNet, there are seven different models, and there is no discernible pattern in the data. Finally, for FDMobileNet, there are three different pre-trained models, and, again, the $\hat{\alpha}$ is larger for the less accurate models. We have not looked in detail at these results and simply present them for completeness.

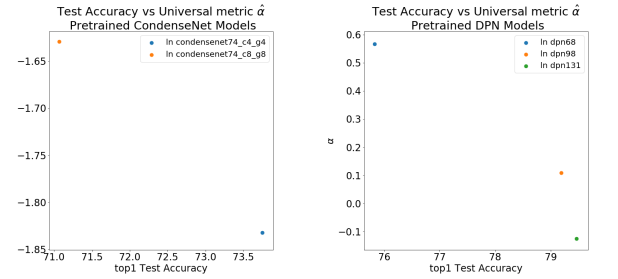
Architecture	Model	Top 1	$\hat{\alpha}$
Working Examples			
DenseNet	densenet121	74.43	1.25
	densenet161	77.14	0.84
	densenet169	75.60	0.68
	densenet201	76.90	0.50
SqueezeNet	squeezenet_v1_0	58.69	2.55
	squeezenet_v1_1	58.18	1.56
CondenseNet	condensenet74_c4_g4	73.75	-1.83
	condensenet74_c8_g8	71.07	-1.63
DPN	dpn68	75.83	0.57
	dpn98	79.19	0.11
	dpn131	79.46	-0.13
ShuffleNet	shufflenetv2_wd2	58.52	5.12
	shufflenetv2_w1	65.61	2.86
MobileNet	mobilenet_wd4	53.74	5.54
	mobilenet_wd2	63.70	4.26
	mobilenet_w3d4	66.46	4.41
	mobilenet_w1	70.14	4.19
	mobilenetv2_wd4	50.28	12.12
	mobilenetv2_wd2	63.46	4.69
	mobilenetv2_w3d4	68.11	4.21
	mobilenetv2_w1	70.69	3.50
SE-ResNet	seresnet50	77.53	-0.35
	seresnet101	78.12	-1.24
	seresnet152	78.52	-1.53
SE-ResNeXt	seresnext50_32x4d	79.00	1.81
	seresnext101_32x4d	80.04	0.76
Counter-examples			
ResNeXt	resnext101_32x4d	78.19	1.22
	resnext101_64x4d	78.96	1.34
MeNet	menet108_8x1_g3	56.08	5.31
	menet128_8x1_g4	56.05	4.46
	menet228_12x1_g3	66.43	4.82
	menet256_12x1_g4	66.59	4.97
	menet348_12x1_g3	69.90	5.74
	menet352_12x1_g8	66.69	4.42
	menet456_24x1_g3	71.60	5.11
FDMobileNet	fdmobilenet_wd4	44.23	6.40
	fdmobilenet_wd2	56.15	7.01
	fdmobilenet_w1	65.30	7.10

Table 3: Results for more pre-trained DNN models. Models provided in the OSMR Sandbox, implemented in pyTorch. Top 1 refers to the Top 1 Accuracy, which 100.0 minus the Top 1 reported error.



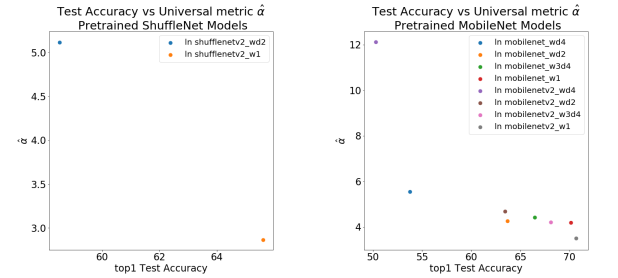
(a) DenseNet (b) SqueezeNet

Figure 3: Pre-trained Densenet and SqueezeNet PyTorch Models. Top 1 Test Accuracy versus $\hat{\alpha}$.



(a) CondenseNet (b) DPN

Figure 4: Pre-trained CondenseNet and DPN Models. Top 1 Test Accuracy versus $\hat{\alpha}$.

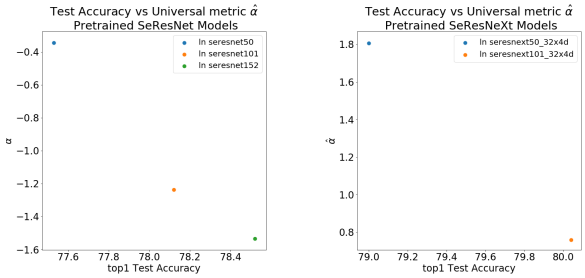


(a) ShuffleNet (b) MobileNet

Figure 5: Pre-trained ShuffleNet and MobileNet Models. Top 1 Test Accuracy versus $\hat{\alpha}$.

5 Discussion and Conclusion

We have presented an *unsupervised* capacity control metric which predicts trends in test accuracies of a trained DNN—without peeking at the test data. See Appendix E of [27] for more discussion. Our work leads to a harder theoretical question: can one characterize properties of realistic DNNs to determine whether a DNN is overtrained—without peeking at the test data?



(a) SeResNet (b) SeResNeXt

Figure 6: Pre-trained SeResNet and SeResNeXt Models. Top 1 Test Accuracy versus $\hat{\alpha}$.

References

[1] A. Swaminathan and T. Joachims. Batch learning from logged bandit feedback through counterfactual risk minimization. *Journal of Machine Learning Research*, 16:1731–1755, 2015.

[2] Q. Liao, B. Miranda, A. Banburski, J. Hidary, and T. Poggio. A surprising linear relationship predicts test performance in deep networks. Technical Report Preprint: arXiv:1807.09659, 2018.

[3] D. Soudry, E. Hoffer, M. S. Nacson, S. Gunasekar, and N. Srebro. The implicit bias of gradient descent on separable data. Technical Report Preprint: arXiv:1710.10345, 2017.

[4] T. Poggio, Q. Liao, B. Miranda, A. Banburski, X. Boix, and J. Hidary. Theory IIIb: Generalization in deep networks. Technical Report Preprint: arXiv:1806.11379, 2018.

[5] B. Neyshabur, R. Tomioka, and N. Srebro. In search of the real inductive bias: on the role of implicit regularization in deep learning. Technical Report Preprint: arXiv:1412.6614, 2014.

[6] B. Neyshabur, R. Tomioka, and N. Srebro. Norm-based capacity control in neural networks. In *Proceedings of the 28th Annual Conference on Learning Theory*, pages 1376–1401, 2015.

[7] B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro. Exploring generalization in deep learning. Technical Report Preprint: arXiv:1706.08947, 2017.

[8] P. Bartlett, D. J. Foster, and M. Telgarsky. Spectrally-normalized margin bounds for neural networks. Technical Report Preprint: arXiv:1706.08498, 2017.

[9] Y. Yoshida and T. Miyato. Spectral norm regularization for improving the generalizability of deep learning. Technical Report Preprint: arXiv:1705.10941, 2017.

[10] K. Kawaguchi, L. P. Kaelbling, and Y. Bengio. Generalization in deep learning. Technical Report Preprint: arXiv:1710.05468, 2017.

[11] B. Neyshabur, S. Bhojanapalli, and N. Srebro. A PAC-Bayesian approach to spectrally-normalized margin

bounds for neural networks. Technical Report Preprint: arXiv:1707.09564, 2017.

[12] S. Arora, R. Ge, B. Neyshabur, and Y. Zhang. Stronger generalization bounds for deep nets via a compression approach. Technical Report Preprint: arXiv:1802.05296, 2018.

[13] S. Arora, N. Cohen, and E. Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. Technical Report Preprint: arXiv:1802.06509, 2018.

[14] P. Zhou and J. Feng. Understanding generalization and optimization performance of deep CNNs. Technical Report Preprint: arXiv:1805.10767, 2018.

[15] P. L. Bartlett. For valid generalization, the size of the weights is more important than the size of the network. In *Annual Advances in Neural Information Processing Systems 9: Proceedings of the 1996 Conference*, pages 134–140, 1997.

[16] M. W. Mahoney and H. Narayanan. Learning with spectral kernels and heavy-tailed data. Technical Report Preprint: arXiv:0906.4539, 2009.

[17] C. H. Martin and M. W. Mahoney. Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning. Technical Report Preprint: arXiv:1810.01075, 2018.

[18] C. H. Martin and M. W. Mahoney. Traditional and heavy-tailed self regularization in neural network models. In *Proceedings of the 36th International Conference on Machine Learning*, pages 4284–4293, 2019.

[19] C. H. Martin and M. W. Mahoney. Unpublished results, 2018.

[20] C. H. Martin and M. W. Mahoney. Rethinking generalization requires revisiting old ideas: statistical mechanics approaches and complex learning behavior. Technical Report Preprint: arXiv:1710.09553, 2017.

[21] D. Sornette. *Critical phenomena in natural sciences: chaos, fractals, selforganization and disorder: concepts and tools*. Springer-Verlag, Berlin, 2006.

[22] J. P. Bouchaud and M. Potters. *Theory of Financial Risk and Derivative Pricing: From Statistical Physics to Risk Management*. Cambridge University Press, 2003.

[23] A. Engel and C. P. L. Van den Broeck. *Statistical mechanics of learning*. Cambridge University Press, New York, NY, USA, 2001.

[24] H. Nishimori. *Statistical Physics of Spin Glasses and Information Processing: An Introduction*. Oxford University Press, Oxford, 2001.

[25] M. Geiger, S. Spigler, S. d'Ascoli, L. Sagun, M. Baity-Jesi, G. Biroli, and M. Wyart. The jamming transition as a paradigm to understand the loss landscape of deep neural networks. Technical Report Preprint: arXiv:1809.09349, 2018.

[26] S. Spigler, M. Geiger, S. d'Ascoli, L. Sagun, G. Biroli, and M. Wyart. A jamming transition from under- to over-parametrization affects loss landscape and generalization. Technical Report Preprint: arXiv:1810.09665, 2018.

[27] C. H. Martin and M. W. Mahoney. Heavy-tailed Universality predicts trends in test accuracies for very large pre-trained deep neural networks. Technical Report Preprint: arXiv:1901.08278v2, 2019.

[28] D. Schleicher. Hausdorff dimension, its properties, and its surprises. *The American Mathematical Monthly*, 114(6):509–528, 2007.