Why Deep Learning Works: Implicit Self-Regularization in Deep Neural Networks

Michael W. Mahoney

ICSI and Dept of Statistics, UC Berkeley

http://www.stat.berkeley.edu/∼mmahoney/

February 2019

(Joint work with Charles H. Martin, Calculation Consulting, charles@calculationconsulting.com)

Perspectives on the talk

- Randomized Numerical Linear Algebra
- Random Matrix Theory
- Foundations of Data Science
- Practical Theory for Learning/Optimization
- Understanding Why Deep Neural Networks Work
- Exploiting Phenomena Like the Generalization Gap
- Engineering Better Learning Algorithms

Outline

1 Background
2 Regularization and the Energy Landscape
3 Preliminary Empirical Results
4 Gaussian and Heavy-tailed Random Matrix Theory
5 More detailed empirical results
6 An RMT-based Theory for Deep Learning
7 Tikhonov regularization versus Heavy-tailed regularization
8 Varying the Batch Size: Explaining the Generalization Gap
9 Applying the Theory
10 Using the Theory
11 More General Implications
12 Conclusions

Motivations: towards a Theory of Deep Learning

Theoretical: deeper insight into *Why Deep Learning Works*?

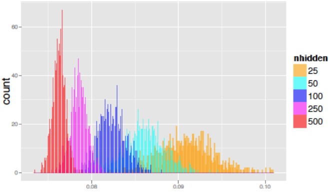
- convex versus non-convex optimization?
- explicit/implicit regularization?
- is / why is / when is deep better?
- VC theory versus Statistical Mechanics theory?
- ...

Practical: use insights to improve engineering of DNNs?

- when is a network fully optimized?
- can we use labels and/or domain knowledge more efficiently?
- large batch versus small batch in optimization?
- designing better ensembles?
- ...

Motivations: towards a Theory of Deep Learning

DNNs as spin glasses, Choromanska et al. 2015



Looks exactly like old protein folding results (late 90s)

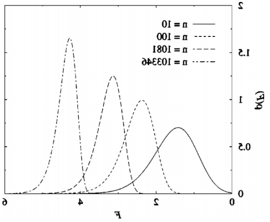
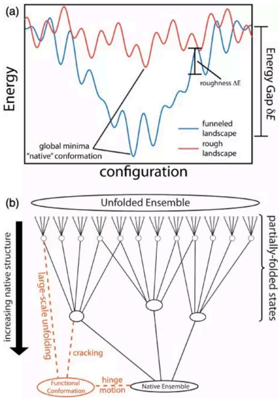


FIG. 1. Plot of the foldability distribution p(F) for different numbers of compact states (n = 10, 100, 1081, 103 346), calculated using the random energy model.

Energy Landscape Theory



Energy Landscape of MultiScale Spin Glass Model

Completely different picture of DNNs

Raises broad questions about Why Deep Learning Works

Set up: the Energy Landscape

Energy/Optimization function:

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

Train this on labeled data $\{d_i, y_i\} \in \mathcal{D}$, using Backprop, by minimizing loss $\mathcal{L}$:

$$\min_{W_l, b_l} \mathcal{L}\left(\sum_i E_{DNN}(d_i) - y_i\right)$$

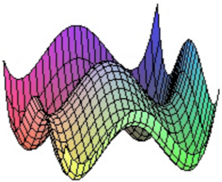
E_{DNN} is "the" *Energy Landscape*:

- The part of the optimization problem parameterized by the heretofore unknown elements of the weight matrices and bias vectors, and as defined by the data $\{d_i, y_i\} \in \mathcal{D}$
- Pass the data through the Energy function E_{DNN} multiple times, as we run Backprop training
- The Energy Landscape* is *changing* at each epoch

*i.e., the optimization function that is *nominally* being optimized

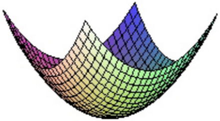
Problem: How can this possibly work?

Expected



Highly non-convex?

Observed

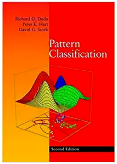


Apparently not!

It has been known for a long time that local minima are not the issue.

Problem: Local Minima?

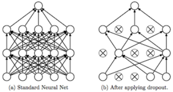
Duda, Hart and Stork, 2000



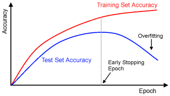
> Whereas in low-dimensional spaces, local minima can be plentiful, in high dimension, the problem of local minima is different: The high-dimensional space may afford more ways (dimensions) for the system to "get around" a barrier or local maximum during learning. The more superfluous the weights, the less likely it is a network will get trapped in local minima. However, networks with an unnecessarily large number of weights are undesirable because of the dangers of overfitting, as we shall see in Section 6.11.

Solution: add more capacity and regularize, i.e., over-parameterization

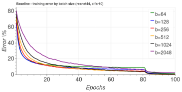
Motivations: what is regularization?



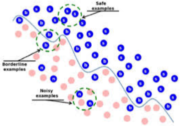
(a) Dropout.



(b) Early Stopping.



(c) Batch Size.



(d) Noisify Data.

Every adjustable *knob* and *switch*—and there are *many*[†]—is regularization.

[†] https://arxiv.org/pdf/1710.10686.pdf

Problem: regularization in DNNs?

ICLR 2017 Best paper

- Large neural network models can easily overtrain/overfit on randomly labeled data
- Popular ways to regularize (basically $\min_x f(x) + \lambda g(x)$) may or may not help.

Understanding deep learning requires rethinking generalization??
`https://arxiv.org/abs/1611.03530`

Rethinking generalization requires revisiting old ideas: statistical mechanics approaches and complex learning behavior!!
`https://arxiv.org/abs/1710.09553`

Outline

1. Background
2. Regularization and the Energy Landscape
3. Preliminary Empirical Results
4. Gaussian and Heavy-tailed Random Matrix Theory
5. More detailed empirical results
6. An RMT-based Theory for Deep Learning
7. Tikhonov regularization versus Heavy-tailed regularization
8. Varying the Batch Size: Explaining the Generalization Gap
9. Applying the Theory
10. Using the Theory
11. More General Implications
12. Conclusions

Basics of Regularization

Ridge Regression / Tikhonov-Phillips Regularization

$$\hat{\mathbf{W}}\mathbf{x} = \mathbf{y}$$
$$\hat{\mathbf{X}} = \hat{\mathbf{W}}^T\hat{\mathbf{W}}$$
$$\mathbf{x} = \left(\hat{\mathbf{X}} + \alpha I\right)^{-1}\hat{\mathbf{W}}^T\mathbf{y}$$

{ *Moore-Penrose pseudoinverse (1955)*
Ridge regularization (Phillips, 1962)

$$\min_{\mathbf{W}_{ij}} \|\hat{\mathbf{W}}\mathbf{x} - \mathbf{y}\|_2^2 + \alpha\|\hat{\mathbf{W}}\|_2^2$$

familiar optimization problem

Softens the rank of **X** to focus on large eigenvalues.

Related to Truncated SVD, which performs hard truncation of rank of **X**

Early stopping, truncated random walks, etc. often implicitly solve regularized optimiation problems.

How we will study regularization

The Energy Landscape is *determined* by layer weight matrices $\mathbf{W}_L$:

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

Traditional regularization is applied to $\mathbf{W}_L$:

$$\min_{W_l, b_l} \mathcal{L}\left(\sum_i E_{DNN}(d_i) - y_i\right) + \alpha \sum_l \|\mathbf{W}_l\|$$

Different types of regularization, e.g., different norms $\|\cdot\|$, *leave different empirical signatures on* $\mathbf{W}_L$.

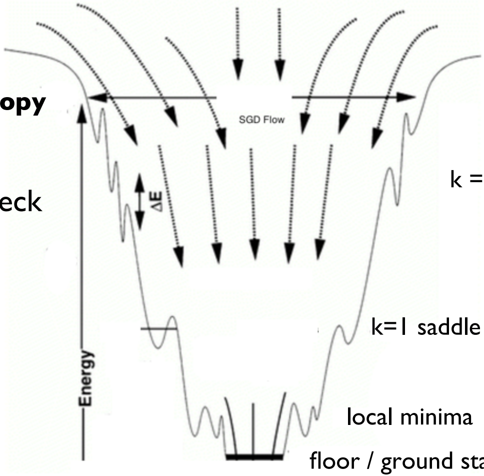
What we do:

- Turn off "all" regularization.
- Systematically turn it back on, explicitly with α or implicitly with knobs/switches.
- Study empirical properties of $\mathbf{W}_L$.

Energy Landscape: and Information flow

Information / Entropy

Information bottleneck
Entropy collapse



k = 2 saddle points

k=1 saddle points

local minima

floor / ground state

Question: What happens to the layer weight matrices $\mathbf{W}_L$?

Lots of DNNs Analyzed

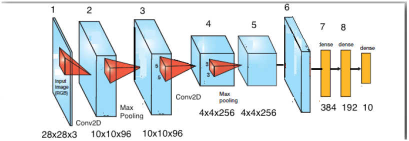
Question: What happens to the layer weight matrices $\mathbf{W}_L$?

(Don't evaluate your method on one/two/three NN, evaluate it on a dozen/hundred.)

Retrained LeNet5 on MINST using Keras.

Two other small models:

- 3-Layer MLP
- Mini AlexNet



Conv2D MaxPool Conv2D MaxPool FC1 FC2 FC

Wide range of state-of-the-art pre-trained models:

- AlexNet, Inception, etc.

Outline

1. Background
2. Regularization and the Energy Landscape
3. Preliminary Empirical Results
4. Gaussian and Heavy-tailed Random Matrix Theory
5. More detailed empirical results
6. An RMT-based Theory for Deep Learning
7. Tikhonov regularization versus Heavy-tailed regularization
8. Varying the Batch Size: Explaining the Generalization Gap
9. Applying the Theory
10. Using the Theory
11. More General Implications
12. Conclusions

A Warmup to *Lots* of DNNs Analyzed

3-Layer MLP:

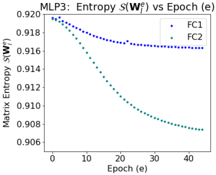
- 3 fully connected (FC) / dense layers with 512 nodes and ReLU activation, with a final FC layer with 10 nodes and softmax activation:

$$
\begin{aligned}
\mathbf{W}_1 &= (\cdot \times 512) \\
\mathbf{W}_2 &= (512 \times 512) \quad \text{(Layer FC1)} \quad (Q = 1) \\
\mathbf{W}_3 &= (512 \times 512) \quad \text{(Layer FC2)} \quad (Q = 1) \\
\mathbf{W}_4 &= (512 \times 10).
\end{aligned}
$$

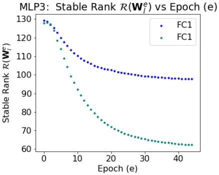
Matrix complexity: Matrix Entropy and Stable Rank

$$\mathbf{W} = \mathbf{U}\Sigma\mathbf{V}^T \qquad \nu_i = \Sigma_{ii} \qquad p_i = \nu_i^2 / \sum_i \nu_i^2$$

$$\mathcal{S}(\mathbf{W}) = \frac{-1}{\log(R(\mathbf{W}))} \sum_i p_i \log\ p_i \qquad \mathcal{R}_s(\mathbf{W}) = \frac{\|\mathbf{W}\|_F^2}{\|\mathbf{W}\|_2^2} = \frac{\sum_i \nu_i^2}{\nu_{max}^2}$$



(e) MLP3 Entropies.



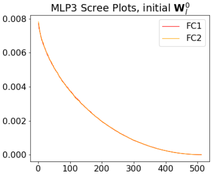
(f) MLP3 Stable Ranks.

Figure: Matrix Entropy & Stable Rank show transition during Backprop training.

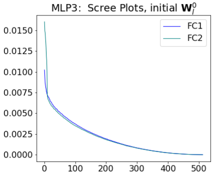
Matrix complexity: Scree Plots

$$\mathbf{W} = \mathbf{U}\Sigma\mathbf{V}^T \qquad \nu_i = \Sigma_{ii} \qquad p_i = \nu_i^2 / \sum_i \nu_i^2$$

$$\mathcal{S}(\mathbf{W}) = \frac{-1}{\log(R(\mathbf{W}))} \sum_i p_i \log \; p_i \qquad \mathcal{R}_s(\mathbf{W}) = \frac{\|\mathbf{W}\|_F^2}{\|\mathbf{W}\|_2^2} = \frac{\sum_i \nu_i^2}{\nu_{max}^2}$$



(a) Initial Scree Plot.



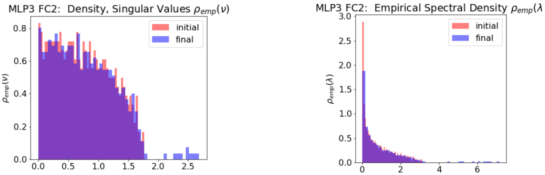
(b) Final Scree Plot.

Figure: Scree plots for initial and final configurations for MLP3.

Matrix complexity: Singular/Eigen Value Densities

$$\mathbf{W} = \mathbf{U}\Sigma\mathbf{V}^T \qquad \nu_i = \Sigma_{ii} \qquad p_i = \nu_i^2 / \sum_i \nu_i^2$$

$$\mathcal{S}(\mathbf{W}) = \frac{-1}{\log(R(\mathbf{W}))} \sum_i p_i \log \ p_i \qquad \mathcal{R}_s(\mathbf{W}) = \frac{\|\mathbf{W}\|_F^2}{\|\mathbf{W}\|_2^2} = \frac{\sum_i \nu_i^2}{\nu_{max}^2}$$



(a) Singular val. density (b) Eigenvalue density

Figure: Histograms of the Singular Values ν_i and associated Eigenvalues $\lambda_i = \nu_i^2$.

ESD: detailed insight into W_L

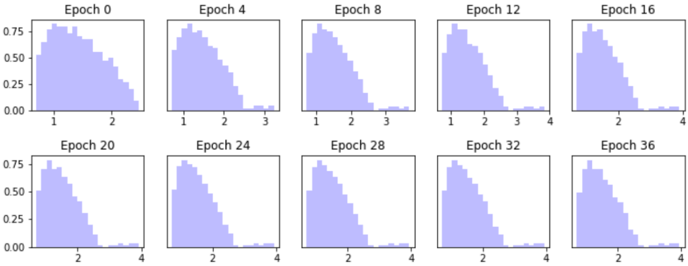
Empirical Spectral Density (ESD: eigenvalues of $X = \mathbf{W}_L^T \mathbf{W}_L$)

```
import keras
import numpy as np
import matplotlib.pyplot as plt
…
W = model.layers[i].get_weights()[0]
…
X = np.dot(W, W.T)
evals, evecs = np.linalg.eig(W, W.T)
plt.hist(X, bin=100, density=True)
```

ESD: detailed insight into W_L

Empirical Spectral Density (ESD: eigenvalues of $X = \mathbf{W}_L^T \mathbf{W}_L$)

Eopch 0: Random Matrix



Eopch 36: Random + Spiles

Entropy decrease corresponds to:

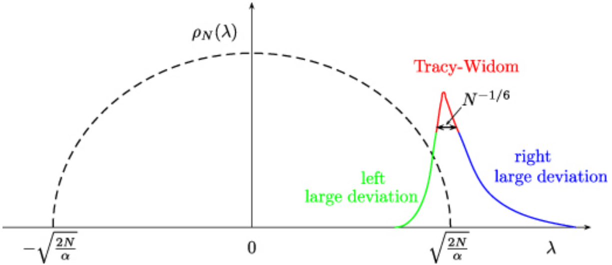
- modification (later, breakdown) of random structure and
- onset of a new kind of self-regularization.

Outline

1 Background
2 Regularization and the Energy Landscape
3 Preliminary Empirical Results
4 Gaussian and Heavy-tailed Random Matrix Theory
5 More detailed empirical results
6 An RMT-based Theory for Deep Learning
7 Tikhonov regularization versus Heavy-tailed regularization
8 Varying the Batch Size: Explaining the Generalization Gap
9 Applying the Theory
10 Using the Theory
11 More General Implications
12 Conclusions

Random Matrix Theory 101: Wigner and Tracy-Widom

- Wigner: *global bulk statistics* approach universal semi-circular form
- Tracy-Widom: *local edge statistics* fluctuate in universal way



Problems with Wigner and Tracy-Widom:

- Weight matrices usually not square
- Typically do only a single training run

Random Matrix Theory 102: Marchenko-Pastur

Let $\mathbf{W}$ be an $N \times M$ random matrix, with elements $W_{ij} \sim N(0, \sigma_{mp}^2)$.

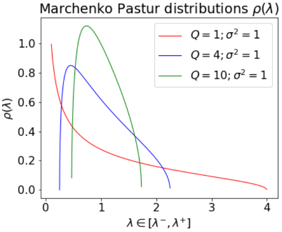
Then, the ESD of $\mathbf{X} = \mathbf{W}^T\mathbf{W}$, converges to a deterministic function:

$$\begin{aligned} \rho_N(\lambda) \quad &:= \quad \frac{1}{N}\sum_{i=1}^{M} \delta\left(\lambda - \lambda_i\right) \\ &\xrightarrow[Q \text{ fixed}]{N\to\infty} \begin{cases} \frac{Q}{2\pi\sigma_{mp}^2}\frac{\sqrt{(\lambda^+ - \lambda)(\lambda - \lambda^-)}}{\lambda} & \text{if } \lambda \in [\lambda^-, \lambda^+] \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

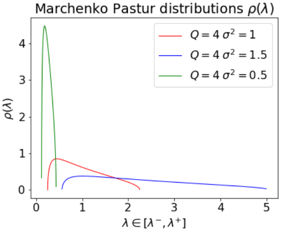
with well-defined edges (which depend on Q, the aspect ratio):

$$\lambda^{\pm} = \sigma_{mp}^2 \left(1 \pm \frac{1}{\sqrt{Q}}\right)^2 \qquad Q = N/M \geq 1.$$

Random Matrix Theory 102': Marchenko-Pastur



(a) Vary aspect ratios



(b) Vary variance parameters

Figure: Marchenko-Pastur (MP) distributions.

Important points:

- *Global bulk stats*: The overall shape is deterministic, fixed by Q and σ.
- *Local edge stats*: The edge λ^+ is very crisp, i.e., $\Delta\lambda_M = |\lambda_{max} - \lambda^+| \sim O(M^{-2/3})$, plus Tracy-Widom fluctuations.

We use both *global bulk statistics* as well as *local edge statistics* in our theory.

Random Matrix Theory 103: Heavy-tailed RMT

Go beyond the (relatively easy) Gaussian Universality class:

- *model* strongly-correlated systems ("signal") with heavy-tailed random matrices.

	Generative Model w/ elements from Universality class	Finite-N Global shape $\rho_N(\lambda)$	Limiting Global shape $\rho(\lambda),\ N \to \infty$	Bulk edge Local stats $\lambda \approx \lambda^+$	(far) Tail Local stats $\lambda \approx \lambda_{max}$
Basic MP	Gaussian	MP distribution	MP	TW	No tail.
Spiked-Covariance	Gaussian, + low-rank perturbations	MP + Gaussian spikes	MP	TW	Gaussian
Heavy tail, $4 < \mu$	(Weakly) Heavy-Tailed	MP + PL tail	MP	Heavy-Tailed*	Heavy-Tailed*
Heavy tail, $2 < \mu < 4$	(Moderately) Heavy-Tailed (or "fat tailed")	PL** $\sim \lambda^{-(a\mu+b)}$	PL $\sim \lambda^{-(\frac{1}{2}\mu+1)}$	No edge.	Frechet
Heavy tail, $0 < \mu < 2$	(Very) Heavy-Tailed	PL** $\sim \lambda^{-(\frac{1}{2}\mu+1)}$	PL $\sim \lambda^{-(\frac{1}{2}\mu+1)}$	No edge.	Frechet

Basic MP theory, and the spiked and Heavy-Tailed extensions we use, including known, empirically-observed, and conjectured relations between them. Boxes marked "*" are best described as following "TW with large finite size corrections" that are likely Heavy-Tailed, leading to bulk edge statistics and far tail statistics that are indistinguishable. Boxes marked "**" are phenomenological fits, describing large ($2 < \mu < 4$) or small ($0 < \mu < 2$) finite-size corrections on $N \to \infty$ behavior.

Fitting Heavy-tailed Distributions

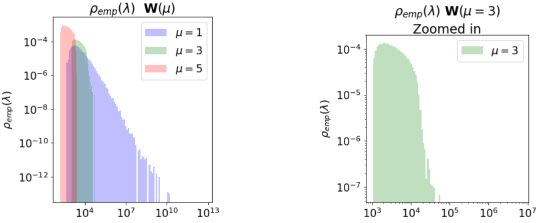
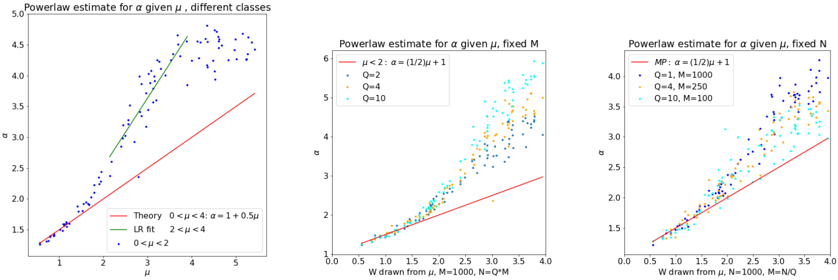


Figure: The log-log histogram plots of the ESD for three Heavy-Tailed random matrices **M** with same aspect ratio $Q = 3$, with $\mu = 1.0, 3.0, 5.0$, corresponding to the three Heavy-Tailed Universality classes ($0 < \mu < 2$ vs $2 < \mu < 4$ and $4 < \mu$).

Non-negligibe finite size effects



(a) $M = 1000, N = 2000$. (b) Fixed M. (c) Fixed N.

Figure: Dependence of α (the fitted PL parameter) on μ (the hypothesized limiting PL parameter).

Outline

1 Background

2 Regularization and the Energy Landscape

3 Preliminary Empirical Results

4 Gaussian and Heavy-tailed Random Matrix Theory

5 More detailed empirical results

6 An RMT-based Theory for Deep Learning

7 Tikhonov regularization versus Heavy-tailed regularization

8 Varying the Batch Size: Explaining the Generalization Gap

9 Applying the Theory

10 Using the Theory

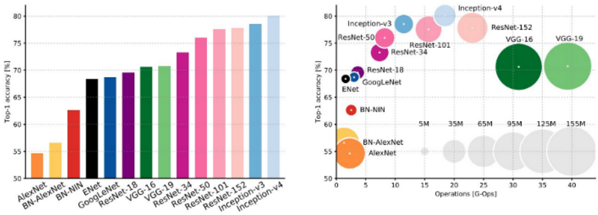
11 More General Implications

12 Conclusions

Experiments: just apply this to pre-trained models

https://medium.com/@siddharthdas_32104/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-...

Year	CNN	Developed by	Place	Top-5 error rate	No. of parameters
1998	LeNet(8)	Yann LeCun et al			60 thousand
2012	AlexNet(7)	Alex Krizhevsky, Geoffrey Hinton, Ilya Sutskever	1st	15.3%	60 million
2013	ZFNet()	Matthew Zeiler and Rob Fergus	1st	14.8%	
2014	GoogLeNet(19)	Google	1st	6.67%	4 million
2014	VGG Net(16)	Simonyan, Zisserman	2nd	7.3%	138 million
2015	ResNet(152)	Kaiming He	1st	3.6%	

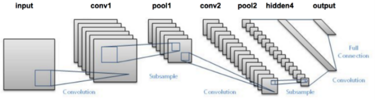


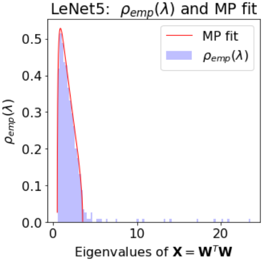
An Analysis of Deep Neural Network Models for Practical Applications, 2017.

Experiments: just apply this to pre-trained models

Model	Layer	Q	$(M \times N)$	α	D	Best Fit
alexnet	17/FC1	2.25	(4096×9216)	2.29	0.0527	PL
	20/FC2	1	(4096×4096)	2.25	0.0372	PL
	22/FC3	4.1	(1000×4096)	3.02	0.0186	PL
densenet121	432	1.02	(1000×1024)	3.32	0.0383	PL
densenet121	432	1.02	(1000×1024)	3.32	0.0383	PL
densenet161	572	2.21	(1000×2208)	3.45	0.0322	PL
densenet169	600	1.66	(1000×1664)	3.38	0.0396	PL
densenet201	712	1.92	(1000×1920)	3.41	0.0332	PL
inception v3	L226	1.3	(768×1000)	5.26	0.0421	PL
	L302	2.05	(1000×2048)	4.48	0.0275	PL
resnet101	286	2.05	(1000×2048)	3.57	0.0278	PL
resnet152	422	2.05	(1000×2048)	3.52	0.0298	PL
resnet18	67	1.95	(512×1000)	3.34	0.0342	PL
resnet34	115	1.95	(512×1000)	3.39	0.0257	PL
resnet50	150	2.05	(1000×2048)	3.54	0.027	PL
vgg11	24	6.12	(4096×25088)	2.32	0.0327	PL
	27	1	(4096×4096)	2.17	0.0309	TPL
	30	4.1	(1000×4096)	2.83	0.0398	PL
vgg11 bn	32	6.12	(4096×25088)	2.07	0.0311	TPL
	35	1	(4096×4096)	1.95	0.0336	TPL
	38	4.1	(1000×4096)	2.99	0.0339	PL
vgg16	34	6.12	(4096×25088)	2.3	0.0277	PL
	37	1	(4096×4096)	2.18	0.0321	TPL
	40	4.1	(1000×4096)	2.09	0.0403	TPL
vgg16 bn	47	6.12	(4096×25088)	2.05	0.0285	TPL
	50	1	(4096×4096)	1.97	0.0363	TPL
	53	4.1	(1000×4096)	3.03	0.0358	PL
vgg19	40	6.12	(4096×25088)	2.27	0.0247	PL
	43	1	(4096×4096)	2.19	0.0313	PL
	46	4.1	(1000×4096)	2.07	0.0368	TPL
vgg19 bn	56	6.12	(4096×25088)	2.04	0.0295	TPL
	59	1	(4096×4096)	1.98	0.0373	TPL
	62	4.1	(1000×4096)	3.03	0.035	PL

RMT: LeNet5 (an old/small example)





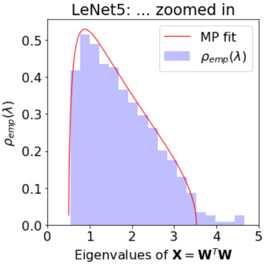
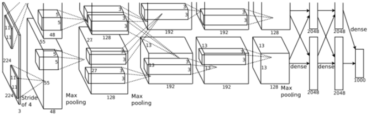


Figure: Full and zoomed-in ESD for LeNet5, Layer FC1.

Marchenko-Pastur Bulk + Spikes

RMT: AlexNet (a typical modern DNN example)



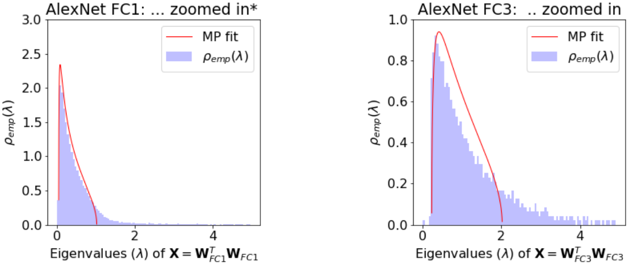
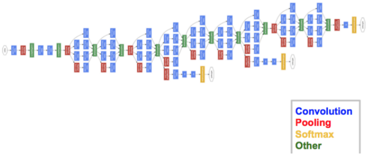


Figure: Zoomed-in ESD for Layer FC1 and FC3 of AlexNet.

Marchenko-Pastur Bulk-decay + Heavy-tailed

RMT: InceptionV3 (a particularly unusual example)



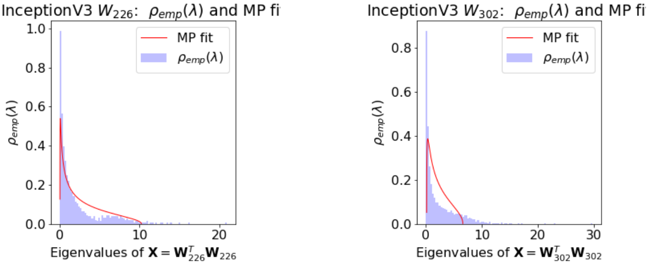


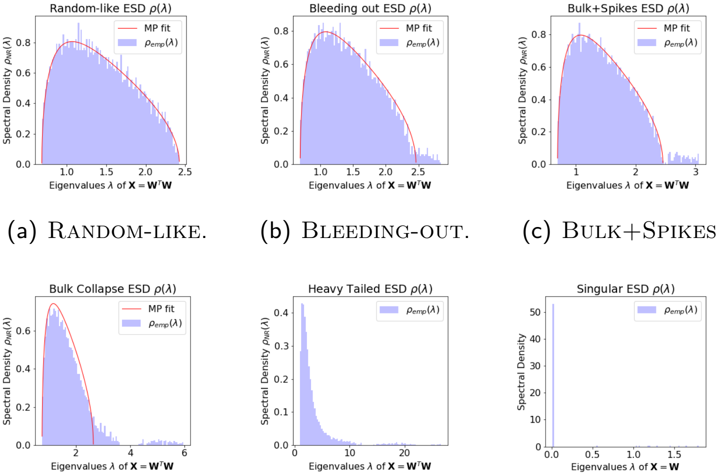
Figure: ESD for Layers L226 and L302 in InceptionV3, as distributed w/ pyTorch.

Marchenko-Pastur bulk decay, onset of Heavy Tails

Outline

1. Background
2. Regularization and the Energy Landscape
3. Preliminary Empirical Results
4. Gaussian and Heavy-tailed Random Matrix Theory
5. More detailed empirical results
6. **An RMT-based Theory for Deep Learning**
7. Tikhonov regularization versus Heavy-tailed regularization
8. Varying the Batch Size: Explaining the Generalization Gap
9. Applying the Theory
10. Using the Theory
11. More General Implications
12. Conclusions

RMT-based 5+1 Phases of Training



(a) Random-like. (b) Bleeding-out. (c) Bulk+Spikes.

(d) Bulk-decay. (e) Heavy-Tailed. (f) Rank-collapse.

Figure: The 5+1 phases of learning we identified in DNN training.

RMT-based 5+1 Phases of Training

We *model* "noise" and also "signal" with random matrices:

$$\mathbf{W} \simeq \mathbf{W}^{rand} + \Delta^{sig}. \quad (1)$$

	Operational Definition	Informal Description via Eqn. (1)	Edge/tail Fluctuation Comments	Illustration and Description
RANDOM-LIKE	ESD well-fit by MP with appropriate λ^+	$\mathbf{W}^{rand}$ random; $\|\Delta^{sig}\|$ zero or small	$\lambda_{max} \approx \lambda^+$ is sharp, with TW statistics	Fig. 10(a)
BLEEDING-OUT	ESD RANDOM-LIKE, excluding eigenmass just above λ^+	$\mathbf{W}$ has eigenmass at bulk edge as spikes "pull out"; $\|\Delta^{sig}\|$ medium	BPP transition, λ_{max} and λ^+ separate	Fig. 10(b)
BULK+SPIKES	ESD RANDOM-LIKE plus ≥ 1 spikes well above λ^+	$\mathbf{W}^{rand}$ well-separated from low-rank Δ^{sig}; $\|\Delta^{sig}\|$ larger	λ^+ is TW, λ_{max} is Gaussian	Fig. 10(c)
BULK-DECAY	ESD less RANDOM-LIKE; Heavy-Tailed eigenmass above λ^+; some spikes	Complex Δ^{sig} with correlations that don't fully enter spike	Edge above λ^+ is not concave	Fig. 10(d)
HEAVY-TAILED	ESD better-described by Heavy-Tailed RMT than Gaussian RMT	$\mathbf{W}^{rand}$ is small; Δ^{sig} is large and strongly-correlated	No good λ^+; $\lambda_{max} \gg \lambda^+$	Fig. 10(e)
RANK-COLLAPSE	ESD has large-mass spike at $\lambda = 0$	$\mathbf{W}$ very rank-deficient; over-regularization	—	Fig. 10(f)

The 5+1 phases of learning we identified in DNN training.

RMT-based 5+1 Phases of Training

Lots of technical issues ...

Outline

1 Background
2 Regularization and the Energy Landscape
3 Preliminary Empirical Results
4 Gaussian and Heavy-tailed Random Matrix Theory
5 More detailed empirical results
6 An RMT-based Theory for Deep Learning
7 Tikhonov regularization versus Heavy-tailed regularization
8 Varying the Batch Size: Explaining the Generalization Gap
9 Applying the Theory
10 Using the Theory
11 More General Implications
12 Conclusions

Bulk+Spikes: Small Models

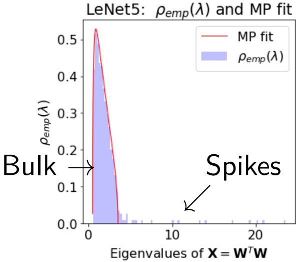
Low-rank perturbation

$$\mathbf{W}_l \simeq \mathbf{W}_l^{rand} + \Delta^{large}$$

Perturbative correction

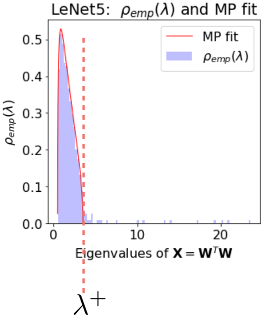
$$\lambda_{max} \;=\; \sigma^2 \left(\frac{1}{Q} + \frac{|\Delta|^2}{N}\right)\left(1 + \frac{N}{|\Delta|^2}\right)$$

$$|\Delta| > (Q)^{-\frac{1}{4}}$$



Smaller, older models can be described perturbatively with Gaussian RMT

Bulk+Spikes: Small Models $\sim$ Tikhonov regularization



simple scale threshold

$$\mathbf{x} = \left(\hat{\mathbf{X}} + \alpha \mathbf{I}\right)^{-1} \hat{\mathbf{W}}^T \mathbf{y}$$

eigenvalues $> \alpha$ (Spikes) carry most of the signal/information

Smaller, older models like LeNet5 exhibit traditional regularization

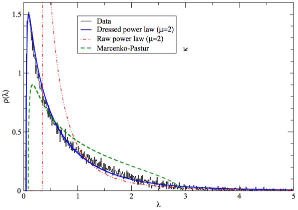
Heavy-tailed Self-regularization

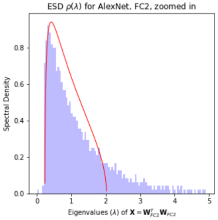
W is *strongly-correlated* and highly non-random

- Can *model* strongly-correlated systems by heavy-tailed random matrices

Then RMT/MP ESD will also have heavy tails

Known results from RMT / polymer theory (Bouchaud, Potters, etc)





AlexNet
ReseNet50
Inception V3
DenseNet201
...

Larger, modern DNNs exhibit novel Heavy-tailed self-regularization

Heavy-tailed Self-regularization

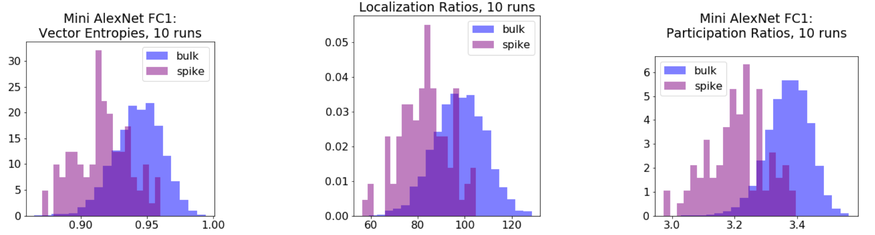
Summary of what we "suspect" today

- No single scale threshold.
- No simple low rank approximation for $\mathbf{W}_L$.
- Contributions from correlations at all scales.
- Can *not* be treated perturbatively.

Larger, modern DNNs exhibit novel Heavy-tailed self-regularization

Spikes: carry more “information” than the Bulk

Spikes have less entropy, are more localized than bulk.



(a) Vector Entropies. (b) Localization Ratios. (c) Participation Ratios.

Figure: Eigenvector localization metrics for the FC1 layer of MiniAlexNet.

Information begins to concentrate in the spikes.

Outline

1. Background
2. Regularization and the Energy Landscape
3. Preliminary Empirical Results
4. Gaussian and Heavy-tailed Random Matrix Theory
5. More detailed empirical results
6. An RMT-based Theory for Deep Learning
7. Tikhonov regularization versus Heavy-tailed regularization
8. Varying the Batch Size: Explaining the Generalization Gap
9. Applying the Theory
10. Using the Theory
11. More General Implications
12. Conclusions

Self-regularization: Batch size experiments

A theory should make predictions:

- We predict the existence of 5+1 phases of increasing implicit self-regularization
- We characterize their properties in terms of HT-RMT

Do these phases exist? Can we find them?

There are *many* knobs. Let's vary one—batch size.

- Tune the batch size from very large to very small
- A small (i.e., retrainable) model exhibits all 5+1 phases
- Large batch sizes => decrease generalization accuracy
- Large batch sizes => decrease implicit self-regularization

Generalization Gap Phenomena: all else being equal, small batch sizes lead to more implicitly self-regularized models.

Batch Size Tuning: Generalization Gap

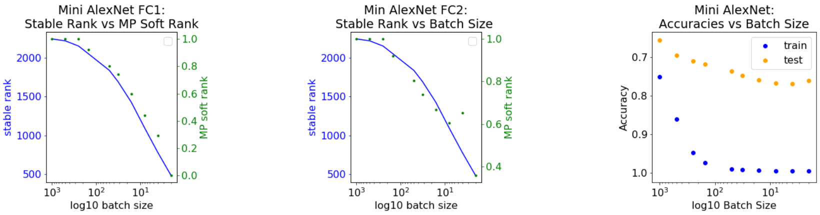
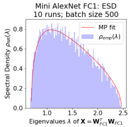


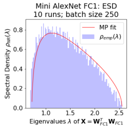
Figure: Varying Batch Size: Stable Rank and MP Softrank for FC1 and FC2 Training and Test Accuracies versus Batch Size for MiniAlexNet.

- *Decreasing batch size leads to better results—it induces strong correlations in* $\mathbf{W}$.
- *Increasing batch size leads to worse results—it washes out strong correlations in* $\mathbf{W}$.

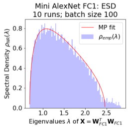
Batch Size Tuning: Generalization Gap



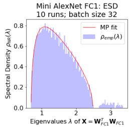
(a) Batch Size 500.



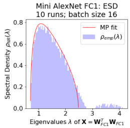
(b) Batch Size 250.



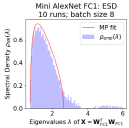
(c) Batch Size 100.



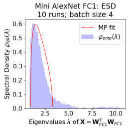
(d) Batch Size 32.



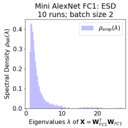
(e) Batch Size 16.



(f) Batch Size 8.



(g) Batch Size 4.



(h) Batch Size 2.

Figure: Varying Batch Size. ESD for Layer FC1 of MiniAlexNet. We exhibit all 5 of the main phases of training by varying only the batch size.

- *Decreasing batch size induces strong correlations in* $\mathbf{W}$*, leading to a more implicitly-regularized model.*
- *Increasing batch size washes out strong correlations in* $\mathbf{W}$*, leading to a less implicitly-regularized model.*

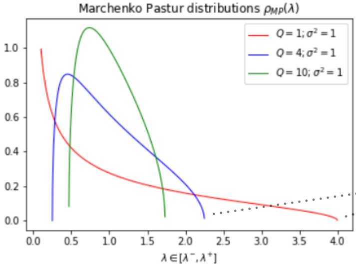
Outline

1 Background
2 Regularization and the Energy Landscape
3 Preliminary Empirical Results
4 Gaussian and Heavy-tailed Random Matrix Theory
5 More detailed empirical results
6 An RMT-based Theory for Deep Learning
7 Tikhonov regularization versus Heavy-tailed regularization
8 Varying the Batch Size: Explaining the Generalization Gap
9 Applying the Theory
10 Using the Theory
11 More General Implications
12 Conclusions

Applying RMT: What phase is your model in?

$Q > 1 \quad : \quad \lambda^{-} > 0$

$Q = 1 \quad : \quad \lambda^{-} = 0$



(a) Different aspect ratios

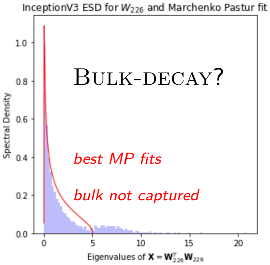
BULK+SPIKES?
BULK-DECAY?
HEAVY-TAILED?

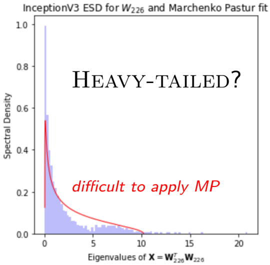
very crisp edges
$\Delta\lambda_M = |\lambda_{max} - \lambda^{+}| \sim O(M^{-2/3})$
plus Tracy-Widom fluctuations

Large, well-trained, modern models approach *Heavy-tailed Self-regularization*.

Applying RMT: What phase is your model in?

Inception V3 Layer 226 $Q \approx 1.3$





Large, well-trained, modern models approach *Heavy-tailed Self-regularization*.

Applying RMT: Heavy Tails $\sim Q = 1$

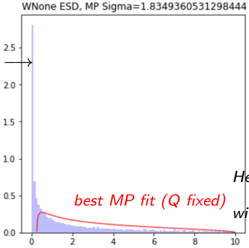
DenseNet201, typical layer, $Q = 1.92$

MP fit is terrible near

eigenvalue minimum $= 0 \longrightarrow$

Resembles Q $= 1$ *fit*

like a soft rank collapse



variance 1.83 is quite large

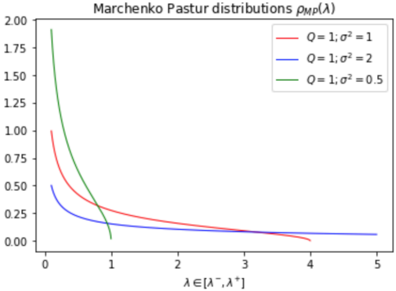
Heavy-tailed, but seemingly

within MP eigenvalue bounds

Large, well-trained, modern models approach *Heavy-tailed Self-regularization*.

Applying RMT: What phase is your model in?

How to apply RMT $Q = 1$ and $\lambda^{-} = 0$



standard MP theory

assumes finite variance

Long tail looks like very large variance

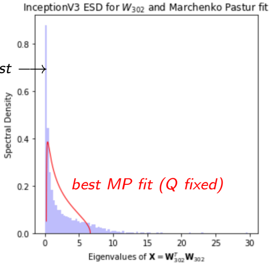
Large, well-trained, modern models approach *Heavy-tailed Self-regularization*.

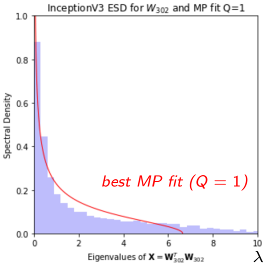
Applying RMT: Should we float Q?

Inception V3 Layer 302 $Q \approx 2.048$

no hard rank

collapse but almost $\longrightarrow$





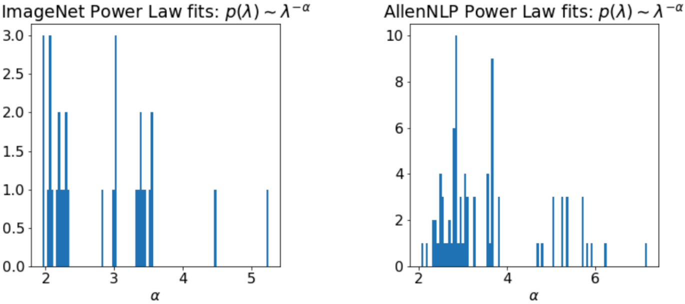
$\lambda_{max} \approx 30$ *(not shown)*

Heavy-tailed, but not clean power law

Heavy-tailed, $Q = 1$ *does not fit*

Large, well-trained, modern models approach *Heavy-tailed Self-regularization*.

Power Law Universality: ImageNet and AllenNLP

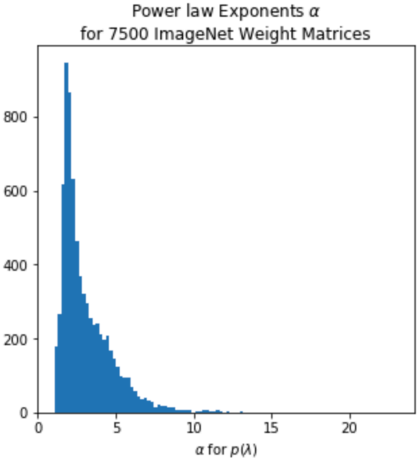


(a) ImageNet pyTorch models　　(b) AllenNLP models

Figure 12: Distribution of power law exponents α for linear layers in pre-trained models trained on ImageNet, available in pyTorch, and for those NLP models, available in AllenNLP.

All these models display remarkable Heavy Tailed Universality

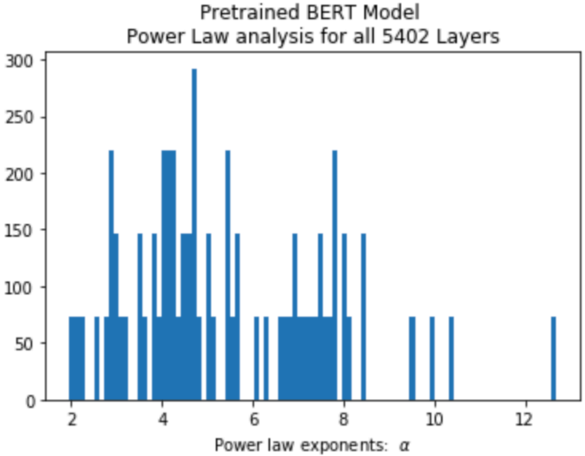
Power Law Universality: ImageNet



- 500 matrices, 50 architectures
- Linear layers and Conv2D feature maps
- $80 - 90\% < 4$

All these models display remarkable Heavy Tailed Universality

Power Law Universality: BERT



The pretrained BERT model is *not* optimal (has large exponents and displays rank collapse)

Summary so far

applied Random Matrix Theory (RMT)

self-regularization $\sim$ entropy / information decrease

5+1 phases of learning

small models $\sim$ Tinkhonov-like regularization

modern DNNs $\sim$ heavy-tailed self-regularization

Remarkably ubiquitous

How can this be used?

Why does deep learning work?

Outline

1. Background
2. Regularization and the Energy Landscape
3. Preliminary Empirical Results
4. Gaussian and Heavy-tailed Random Matrix Theory
5. More detailed empirical results
6. An RMT-based Theory for Deep Learning
7. Tikhonov regularization versus Heavy-tailed regularization
8. Varying the Batch Size: Explaining the Generalization Gap
9. Applying the Theory
10. Using the Theory
11. More General Implications
12. Conclusions

DNN Capacity metrics: Product norms

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

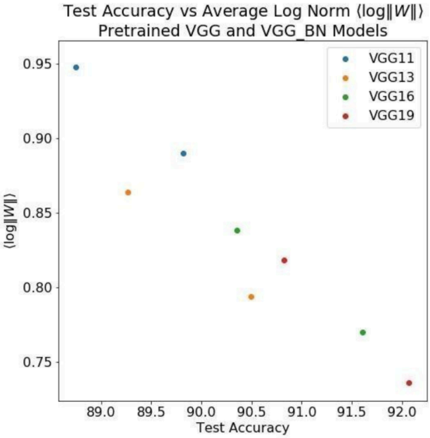
$$\log \mathcal{C} \sim \log \left[\|\mathbf{W}_1\| \times \|\mathbf{W}_2\| \cdots \|\mathbf{W}_L\| \right]$$

$$\log \mathcal{C} \sim \left[\log \|\mathbf{W}_1\| + \log \|\mathbf{W}_2\| \cdots \log \|\mathbf{W}_L\| \right]$$

$$\langle \log \|\mathbf{W}\|_F \rangle = \frac{1}{N_L} \sum_L \log \|\mathbf{W}_L\|$$

The product norm is a VC-like data-dependent capacity metric for DNNs

Predicting test accuracies: Product norms



We can predict trends in the test accuracy—*without peeking at the test data!*

"pip install weightwatcher"

Universality and Capacity control metrics

"Universality" *suggests* the power law exponent α would make a good, Universal, DNN capacity control metric

Imagine a weighted average

$$\hat{\alpha} = \frac{1}{N} \sum_{l,i} b_{l,i} \alpha_{l,i}$$

where the weights b are related to the scale of the weight matrix

This is an *unsupervised* VC-like data-dependent complexity metric for predicting *trends* in average case generalization accuracy in DNNs

- What are the weights $b_{l,i}$?
- We need a relation between the Frobenius norm and the Power Law exponent.

Heavy Tailed matrices: norm-powerlaw relations

- Create a random Heavy Tailed (Pareto) matrix:

$$\Pr\left(W^{rand}_{i,j}\right) \sim \frac{1}{x^{1+\mu}}$$

- Examine the norm-powerlaw relations:

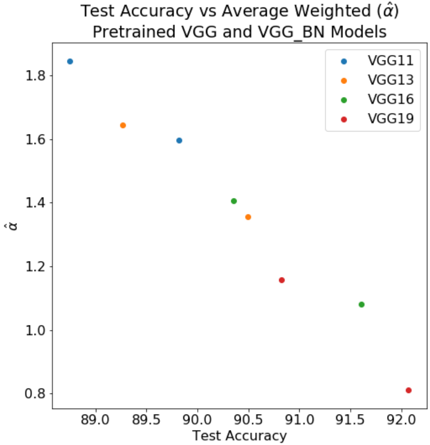
$$\frac{\log \|\mathbf{W}\|_F^2}{\log \lambda_{max}} \quad \text{versus} \quad \alpha$$

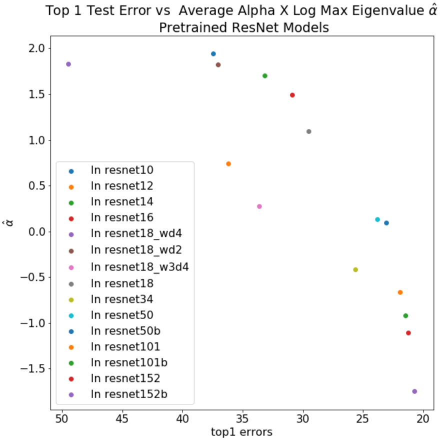
- Argue that:

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

- The weights compensate for different size and scale weight matrices and feature maps.
- Can treat both Linear layers and Conv2D feature maps.

Predicting test accuracies: Weighted Power Laws





We can predict trends in the test accuracy—*without peeking at the test data!*

"pip install weightwatcher"

Outline

1 Background

2 Regularization and the Energy Landscape

3 Preliminary Empirical Results

4 Gaussian and Heavy-tailed Random Matrix Theory

5 More detailed empirical results

6 An RMT-based Theory for Deep Learning

7 Tikhonov regularization versus Heavy-tailed regularization

8 Varying the Batch Size: Explaining the Generalization Gap

9 Applying the Theory

10 Using the Theory

11 More General Implications

12 Conclusions

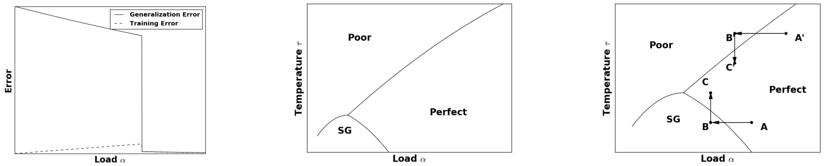
Rethinking generalization requires revisiting old ideas

Martin and Mahoney https://arxiv.org/abs/1710.09553

Very Simple Deep Learning (VSDL) model:

- DNN is a black box, load-like parameters α, & temperature-like parameters τ
- Adding noise to training data decreases α
- Early stopping increases τ

Nearly any non-trivial model[‡] exhibits "phase diagrams," with *qualitatively* different generalization properties, for different parameter values.



(e) Training/generalization error in the VSDL model.

(f) Learning phases in τ-α plane for VSDL model.

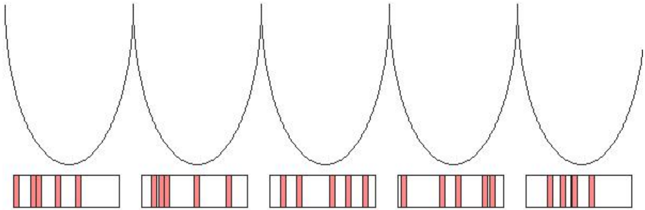
(g) Noisifying data and adjusting knobs.

[‡] *when analyzed via the Statistical Mechanics Theory of Generalization (SMToG)*

Remembering Regularization

Martin and Mahoney https://arxiv.org/abs/1710.09553

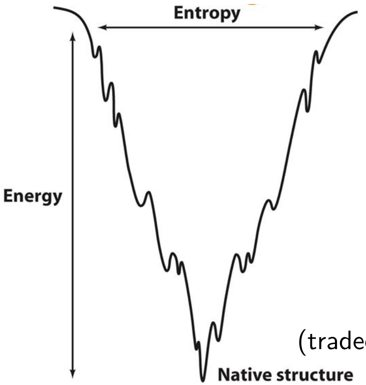
Statistical Mechanics (1990s): (this) Overtraining $\rightarrow$ Spin Glass Phase



Binary Classifier with N Random Labelings:

2^N over-trained solutions: locally (ruggedly) convex, very high barriers, all unable to generalize

Implications: RMT and Deep Learning



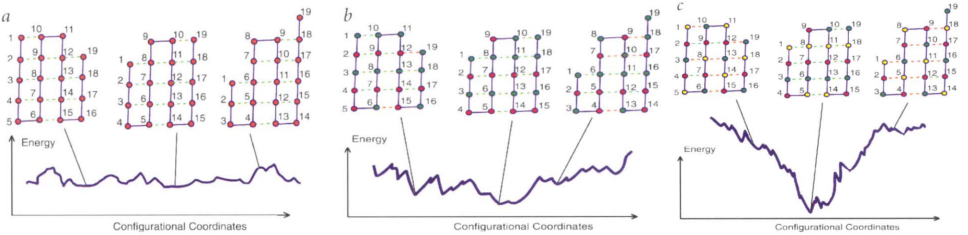
- Where are the local minima?
- How is the Hessian behaved?
- Are simpler models misleading?
- Can we design better learning strategies?

(tradeoff between Energy and Entropy minimization)

How can RMT be used to understand the Energy Landscape?

Implications: Minimizing Frustration and Energy Funnels

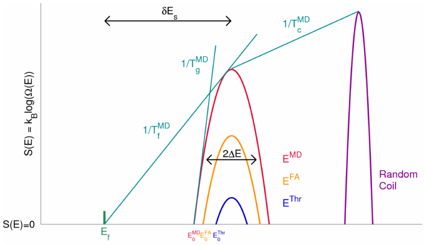
As simple as can be?, Wolynes, 1997

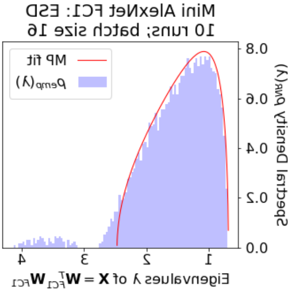


Energy Landscape Theory: "random heteropolymer" versus "natural protein" folding

Implications: The Spin Glass of Minimal Frustration

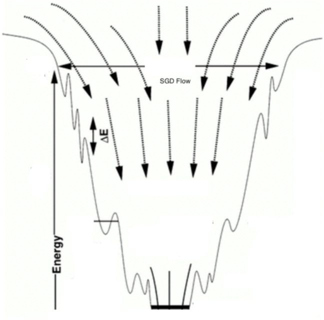
https://calculatedcontent.com/2015/03/25/why-does-deep-learning-work/





low lying Energy state in Spin Glass $\sim$ spikes in RMT

Implications: Energy Landscapes of Heavy-tailed Models?



Compare with (Gaussian) Spin Glass model of Choromanska et al. 2015

Spin Glasses with Heavy Tails?

- Local minima do not concentrate near the ground state (Cizeau and Bouchaud 1993)

If Energy Landscape is more funneled, then no "problems" with local minima!

Outline

1 Background
2 Regularization and the Energy Landscape
3 Preliminary Empirical Results
4 Gaussian and Heavy-tailed Random Matrix Theory
5 More detailed empirical results
6 An RMT-based Theory for Deep Learning
7 Tikhonov regularization versus Heavy-tailed regularization
8 Varying the Batch Size: Explaining the Generalization Gap
9 Applying the Theory
10 Using the Theory
11 More General Implications
12 Conclusions

Finish with the Conclusions

Main Empirical Results:

- Small/old NNs: Tikhonov-like self-regularization
- Modern DNNs: Heavy-tailed self-regularization

Main Modeling Results: $\mathbf{W} \simeq \mathbf{W}^{rand} + \Delta^{sig}$:

- Small/old NNs: model "noise" $\mathbf{W}^{rand}$ with Gaussian random matrices
- Modern DNNs: model strongly-correlated "signal" Δ^{sig} with Heavy-tailed random matrices

Main Theoretical Results: Use Heavy-tailed RMT to:

- Using global bulk stats and local edge stats, construct a operational/phenomenological theory of DNN learning
- Hypothesize 5+1 phases of learning

Evaluating the Theory:

- Effect of implicit versus explicit regularization
- Exhibit all 5+1 phases by adjusting batch size: explain the generalization gap

Main Methodological Contribution:

- Observations → Hypotheses → Build a Theory → Test the Theory.

Many Implications:

- E.g., justify claims about rugged convexity of Energy Landscape

If you want more ...

Background paper:

- Rethinking generalization requires revisiting old ideas: statistical mechanics approaches and complex learning behavior
  (`https://arxiv.org/abs/1710.09553`)

Main paper (full):

- Implicit Self-Regularization in Deep Neural Networks: Evidence from Random Matrix Theory and Implications for Learning
  (`https://arxiv.org/abs/1810.01075`)
- Code: `https://github.com/CalculatedContent/ImplicitSelfRegularization`

Main paper (abridged):

- Traditional and Heavy-Tailed Self Regularization in Neural Network Models
  (`https://arxiv.org/abs/1901.08276`)
- Code: `https://github.com/CalculatedContent/ImplicitSelfRegularization`

Applying the theory paper:

- Heavy-Tailed Universality Predicts Trends in Test Accuracies for Very Large Pre-Trained Deep Neural Networks
  (`https://arxiv.org/abs/1901.08278`)
- Code: `https://github.com/CalculatedContent/PredictingTestAccuracies`
- `https://github.com/CalculatedContent/WeightWatcher`
- "pip install weightwatcher"