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
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)

Energy Landscape Theory

Competeely different picture of DNNs

Raises broad questions about Why Deep Learning Works
Set up: the Energy Landscape

Energy/Optimization function:

\[ E_{DNN} = h_L(W_L \times h_{L-1}(W_{L-1} \times h_{L-2}(...)) + b_{L-1}) + b_L \]

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

\[
\min_{W_i;b_i} \mathcal{L} \left( \sum_i E_{DNN}(d_i) - y_i \right)
\]

\( E_{DNN} \) 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 \( \{d_i, y_i\} \in D \)

- 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?

It has been known for a long time that local minima are not the issue.
Problem: Local Minima?

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.

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

\( \hat{W}x = y \)
\( \hat{X} = \hat{W}^T \hat{W} \)
\( x = (\hat{X} + \alpha I)^{-1} \hat{W}^T y \) \{ Moore-Penrose pseudoinverse (1955) \}
\( \text{Ridge regularization (Phillips, 1962)} \)

\[
\min_{W_{ij}} \left\| \hat{W}x - y \right\|^2_2 + \alpha \left\| \hat{W} \right\|^2_2
\]

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 optimization problems.
How we will study regularization

The Energy Landscape is determined by layer weight matrices $W_L$:

$$E_{DNN} = h_L(W_L \times h_{L-1}(W_{L-1} \times h_{L-2}(\cdots) + b_{L-1}) + b_L)$$

Traditional regularization is applied to $W_L$:

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

Different types of regularization, e.g., different norms $\| \cdot \|$, leave different empirical signatures on $W_L$.

What we do:

- Turn off “all” regularization.
- Systematically turn it back on, explicitly with $\alpha$ or implicitly with knobs/switches.
- Study empirical properties of $W_L$. 
Energy Landscape: and Information flow

Information / Entropy

Information bottleneck
Entropy collapse

local minima
k=1 saddle points

floor / ground state
k = 2 saddle points

Question: What happens to the layer weight matrices $W_L$?
Lots of DNNs Analyzed

**Question:** What happens to the layer weight matrices $W_L$?

*(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

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:

\[
W_1 = (\cdot \times 512) \\
W_2 = (512 \times 512) \quad \text{(Layer FC1) } (Q = 1) \\
W_3 = (512 \times 512) \quad \text{(Layer FC2) } (Q = 1) \\
W_4 = (512 \times 10).
\]
Matrix complexity: Matrix Entropy and Stable Rank

\[ W = U \Sigma V^T \]

\[ \nu_i = \sum_{ij} \]

\[ p_i = \frac{\nu_i^2}{\sum_i \nu_i^2} \]

\[ S(W) = \frac{-1}{\log(R(W))} \sum_i p_i \log p_i \]

\[ R_s(W) = \frac{\|W\|_F^2}{\|W\|_2^2} = \sum_i \frac{\nu_i^2}{\nu_{max}^2} \]

(e) MLP3 Entropies. (f) MLP3 Stable Ranks.

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

\[ W = U \Sigma V^T \quad \nu_i = \sum_{ii} \]

\[ S(W) = \frac{-1}{\log(R(W))} \sum_i p_i \log p_i \]

\[ R_s(W) = \frac{||W||_F^2}{||W||_2^2} = \frac{\sum_i \nu_i^2}{\nu_{\max}^2} \]

(a) Initial Scree Plot. 
(b) Final Scree Plot.

**Figure:** Scree plots for initial and final configurations for MLP3.
Matrix complexity: Singular/Eigen Value Densities

\[ W = U \Sigma V^T \]

\[ \nu_i = \Sigma_{ii} \]

\[ p_i = \nu_i^2 / \sum_i \nu_i^2 \]

\[ S(W) = \frac{-1}{\log(R(W))} \sum_i p_i \log p_i \]

\[ R_s(W) = \frac{\|W\|_F^2}{\|W\|_2^2} = \frac{\sum_i \nu_i^2}{\nu_{\text{max}}^2} \]

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

(a) Singular val. density  
(b) Eigenvalue density
ESD: detailed insight into $W_L$

Empirical Spectral Density (ESD: eigenvalues of $X = W_L^T W_L$)

```python
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 $X = W_L^T W_L$)

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
Let $\mathbf{W}$ be an $N \times M$ random matrix, with elements $W_{ij} \sim N(0, \sigma^2_{mp})$. Then, the ESD of $\mathbf{X} = \mathbf{W}^T \mathbf{W}$, converges to a deterministic function:

$$\rho_N(\lambda) := \frac{1}{N} \sum_{i=1}^{M} \delta(\lambda - \lambda_i)$$

$$\frac{N \to \infty}{Q \text{ fixed}} \left\{ \begin{array}{ll}
Q & \frac{\sqrt{(\lambda^+ - \lambda)(\lambda - \lambda^-)}}{2\pi \sigma^2_{mp}} \frac{\lambda}{\lambda} \\
0 & \text{if } \lambda \in [\lambda^-, \lambda^+] \\
& \text{otherwise.}
\end{array} \right.$$ 

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

$$\lambda^\pm = \sigma^2_{mp} \left(1 \pm \frac{1}{\sqrt{Q}}\right)^2 \quad Q = \frac{N}{M} \geq 1.$$
Random Matrix Theory 102’: Marchenko-Pastur

**Figure:** Marchenko-Pastur (MP) distributions.

(a) Vary aspect ratios

(b) Vary variance parameters

**Important points:**

- **Global bulk stats:** The overall shape is deterministic, fixed by $Q$ and $\sigma$.

- **Local edge stats:** The edge $\lambda^+$ is very crisp, i.e.,

  \[ \Delta \lambda_M = |\lambda_{\max} - \lambda^+| \sim O(M^{-2/3}) , \text{ plus Tracy-Widom fluctuations}. \]

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.

<table>
<thead>
<tr>
<th>Generative Model w/ elements from Universality class</th>
<th>Finite-N Global shape $\rho_N(\lambda)$</th>
<th>Limiting Global shape $\rho(\lambda), N \to \infty$</th>
<th>Bulk edge Local stats $\lambda \approx \lambda^+$</th>
<th>(far) Tail Local stats $\lambda \approx \lambda_{\text{max}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic MP</td>
<td>Gaussian</td>
<td>MP distribution</td>
<td>MP</td>
<td>TW</td>
</tr>
<tr>
<td>Spiked-Covariance</td>
<td>Gaussian, + low-rank perturbations</td>
<td>MP + Gaussian spikes</td>
<td>MP</td>
<td>TW</td>
</tr>
<tr>
<td>Heavy tail, $4 \leq \mu$</td>
<td>(Weakly) Heavy-Tailed</td>
<td>MP + PL tail</td>
<td>MP</td>
<td>Heavy-Tailed*</td>
</tr>
<tr>
<td>Heavy tail, $2 &lt; \mu &lt; 4$</td>
<td>(Moderately) Heavy-Tailed (or “fat tailed”)</td>
<td>PL** $\sim \lambda^{-(a\mu+b)}$</td>
<td>PL $\sim \lambda^{-\left(\frac{1}{2}\mu+1\right)}$</td>
<td>No edge.</td>
</tr>
<tr>
<td>Heavy tail, $0 &lt; \mu &lt; 2$</td>
<td>(Very) Heavy-Tailed</td>
<td>PL** $\sim \lambda^{-(\frac{1}{2}\mu+1)}$</td>
<td>PL $\sim \lambda^{-\left(\frac{1}{2}\mu+1\right)}$</td>
<td>No edge.</td>
</tr>
</tbody>
</table>

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-negligible finite size effects

(a) $M = 1000$, $N = 2000$.

(b) Fixed $M$.

(c) Fixed $N$.

Figure: Dependence of $\alpha$ (the fitted PL parameter) on $\mu$ (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-...
Experiments: just apply this to pre-trained models

<table>
<thead>
<tr>
<th>Model</th>
<th>Layer</th>
<th>Q</th>
<th>( (M \times N) )</th>
<th>( \alpha )</th>
<th>D</th>
<th>Best Fit</th>
</tr>
</thead>
<tbody>
<tr>
<td>alexnet</td>
<td>17/FC1</td>
<td>2.25</td>
<td>( (4096 \times 9216) )</td>
<td>2.29</td>
<td>0.0527</td>
<td>PL</td>
</tr>
<tr>
<td></td>
<td>20/FC2</td>
<td>1</td>
<td>( (4096 \times 4096) )</td>
<td>2.25</td>
<td>0.0372</td>
<td>PL</td>
</tr>
<tr>
<td></td>
<td>22/FC3</td>
<td>4.1</td>
<td>( (1000 \times 1920) )</td>
<td>3.02</td>
<td>0.0186</td>
<td>PL</td>
</tr>
<tr>
<td>densenet121</td>
<td>432</td>
<td>1.02</td>
<td>( (1000 \times 1024) )</td>
<td>3.32</td>
<td>0.0383</td>
<td>PL</td>
</tr>
<tr>
<td>densenet121</td>
<td>432</td>
<td>1.02</td>
<td>( (1000 \times 1024) )</td>
<td>3.32</td>
<td>0.0383</td>
<td>PL</td>
</tr>
<tr>
<td>densenet161</td>
<td>572</td>
<td>2.21</td>
<td>( (1000 \times 1024) )</td>
<td>3.45</td>
<td>0.0322</td>
<td>PL</td>
</tr>
<tr>
<td>densenet169</td>
<td>600</td>
<td>1.66</td>
<td>( (1000 \times 1654) )</td>
<td>3.38</td>
<td>0.0396</td>
<td>PL</td>
</tr>
<tr>
<td>densenet201</td>
<td>712</td>
<td>1.92</td>
<td>( (1000 \times 1920) )</td>
<td>3.41</td>
<td>0.0332</td>
<td>PL</td>
</tr>
<tr>
<td>inception v3</td>
<td>L226</td>
<td>1.3</td>
<td>( (768 \times 1000) )</td>
<td>5.26</td>
<td>0.0421</td>
<td>PL</td>
</tr>
<tr>
<td></td>
<td>L302</td>
<td>2.05</td>
<td>( (1000 \times 2048) )</td>
<td>4.48</td>
<td>0.0275</td>
<td>PL</td>
</tr>
<tr>
<td>resnet101</td>
<td>286</td>
<td>2.05</td>
<td>( (1000 \times 1024) )</td>
<td>3.57</td>
<td>0.0278</td>
<td>PL</td>
</tr>
<tr>
<td>resnet152</td>
<td>422</td>
<td>2.05</td>
<td>( (1000 \times 1024) )</td>
<td>3.52</td>
<td>0.0298</td>
<td>PL</td>
</tr>
<tr>
<td>resnet18</td>
<td>67</td>
<td>1.95</td>
<td>( (512 \times 1000) )</td>
<td>3.34</td>
<td>0.0342</td>
<td>PL</td>
</tr>
<tr>
<td>resnet34</td>
<td>115</td>
<td>1.95</td>
<td>( (512 \times 1000) )</td>
<td>3.39</td>
<td>0.0257</td>
<td>PL</td>
</tr>
<tr>
<td>resnet50</td>
<td>150</td>
<td>2.05</td>
<td>( (1000 \times 1024) )</td>
<td>3.54</td>
<td>0.027</td>
<td>PL</td>
</tr>
<tr>
<td>vgg11</td>
<td>24</td>
<td>6.12</td>
<td>( (4096 \times 25088) )</td>
<td>2.32</td>
<td>0.0327</td>
<td>PL</td>
</tr>
<tr>
<td></td>
<td>27</td>
<td>1</td>
<td>( (4096 \times 4096) )</td>
<td>2.17</td>
<td>0.0309</td>
<td>TPL</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>4.1</td>
<td>( (1000 \times 4096) )</td>
<td>2.83</td>
<td>0.0398</td>
<td>PL</td>
</tr>
<tr>
<td>vgg11 bn</td>
<td>32</td>
<td>6.12</td>
<td>( (4096 \times 25088) )</td>
<td>2.07</td>
<td>0.0311</td>
<td>TPL</td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>1</td>
<td>( (4096 \times 4096) )</td>
<td>1.95</td>
<td>0.0336</td>
<td>TPL</td>
</tr>
<tr>
<td></td>
<td>38</td>
<td>4.1</td>
<td>( (1000 \times 4096) )</td>
<td>2.99</td>
<td>0.0339</td>
<td>PL</td>
</tr>
<tr>
<td>vgg16</td>
<td>34</td>
<td>6.12</td>
<td>( (4096 \times 25088) )</td>
<td>2.3</td>
<td>0.0277</td>
<td>PL</td>
</tr>
<tr>
<td></td>
<td>37</td>
<td>1</td>
<td>( (4096 \times 4096) )</td>
<td>2.18</td>
<td>0.0321</td>
<td>TPL</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>4.1</td>
<td>( (1000 \times 4096) )</td>
<td>2.09</td>
<td>0.0403</td>
<td>TPL</td>
</tr>
<tr>
<td>vgg16 bn</td>
<td>47</td>
<td>6.12</td>
<td>( (4096 \times 25088) )</td>
<td>2.05</td>
<td>0.0285</td>
<td>TPL</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>1</td>
<td>( (4096 \times 4096) )</td>
<td>1.97</td>
<td>0.0363</td>
<td>TPL</td>
</tr>
<tr>
<td></td>
<td>53</td>
<td>4.1</td>
<td>( (1000 \times 4096) )</td>
<td>3.03</td>
<td>0.0358</td>
<td>PL</td>
</tr>
<tr>
<td>vgg19</td>
<td>40</td>
<td>6.12</td>
<td>( (4096 \times 25088) )</td>
<td>2.27</td>
<td>0.0247</td>
<td>PL</td>
</tr>
<tr>
<td></td>
<td>43</td>
<td>1</td>
<td>( (4096 \times 4096) )</td>
<td>2.19</td>
<td>0.0313</td>
<td>PL</td>
</tr>
<tr>
<td></td>
<td>46</td>
<td>4.1</td>
<td>( (1000 \times 4096) )</td>
<td>2.07</td>
<td>0.0368</td>
<td>TPL</td>
</tr>
<tr>
<td>vgg19 bn</td>
<td>56</td>
<td>6.12</td>
<td>( (4096 \times 25088) )</td>
<td>2.04</td>
<td>0.0295</td>
<td>TPL</td>
</tr>
<tr>
<td></td>
<td>59</td>
<td>1</td>
<td>( (4096 \times 4096) )</td>
<td>1.98</td>
<td>0.0373</td>
<td>TPL</td>
</tr>
<tr>
<td></td>
<td>62</td>
<td>4.1</td>
<td>( (1000 \times 4096) )</td>
<td>3.03</td>
<td>0.035</td>
<td>PL</td>
</tr>
</tbody>
</table>
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:

$$W \sim W^{\text{rand}} + \Delta^{\text{sig}}. \quad (1)$$

<table>
<thead>
<tr>
<th>Phase</th>
<th>Operational Definition</th>
<th>Informal Description via Eqn. (1)</th>
<th>Edge/tail Fluctuation Comments</th>
<th>Illustration and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Random-like</strong></td>
<td>ESD well-fit by MP with appropriate $\lambda^+$</td>
<td>$W^{\text{rand}}$ random; $|\Delta^{\text{sig}}|$ zero or small</td>
<td>$\lambda_{\text{max}} \approx \lambda^+$ is sharp, with TW statistics</td>
<td>Fig. 10(a)</td>
</tr>
<tr>
<td><strong>Bleeding-out</strong></td>
<td>ESD Random-like, excluding eigenmass just above $\lambda^+$</td>
<td>$W$ has eigenmass at bulk edge as spikes “pull out”; $|\Delta^{\text{sig}}|$ medium</td>
<td>BPP transition, $\lambda_{\text{max}}$ and $\lambda^+$ separate</td>
<td>Fig. 10(b)</td>
</tr>
<tr>
<td><strong>Bulk+Spires</strong></td>
<td>ESD Random-like plus $\geq 1$ spikes well above $\lambda^+$</td>
<td>$W^{\text{rand}}$ well-separated from low-rank $\Delta^{\text{sig}}$; $|\Delta^{\text{sig}}|$ larger</td>
<td>$\lambda^+$ is TW, $\lambda_{\text{max}}$ is Gaussian</td>
<td>Fig. 10(c)</td>
</tr>
<tr>
<td><strong>Bulk-decay</strong></td>
<td>ESD less Random-like; Heavy-Tailed eigenmass above $\lambda^+$; some spikes</td>
<td>Complex $\Delta^{\text{sig}}$ with correlations that don’t fully enter spike</td>
<td>Edge above $\lambda^+$ is not concave</td>
<td>Fig. 10(d)</td>
</tr>
<tr>
<td><strong>Heavy-Tailed</strong></td>
<td>ESD better-described by Heavy-Tailed RMT than Gaussian RMT</td>
<td>$W^{\text{rand}}$ is small; $\Delta^{\text{sig}}$ is large and strongly-correlated</td>
<td>No good $\lambda^+$; $\lambda_{\text{max}} \gg \lambda^+$</td>
<td>Fig. 10(e)</td>
</tr>
<tr>
<td><strong>Rank-collapse</strong></td>
<td>ESD has large-mass spike at $\lambda = 0$</td>
<td>$W$ very rank-deficient; over-regularization</td>
<td>—</td>
<td>Fig. 10(f)</td>
</tr>
</tbody>
</table>

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

\[ \mathbf{W}_l \simeq \mathbf{W}_l^{\text{rand}} + \Delta^{\text{large}} \]

Perturbative correction

\[ \lambda_{\text{max}} = \sigma^2 \left( \frac{1}{Q} + \frac{\vert\Delta\vert^2}{N} \right) \left( 1 + \frac{N}{\vert\Delta\vert^2} \right) \]

\[ \vert\Delta\vert > (Q)^{-\frac{1}{4}} \]

Smaller, older models can be described perturbatively with Gaussian RMT
Bulk + Spikes: Small Models ∼ Tikhonov regularization

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

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)

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 $W_L$.
- 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 $\Rightarrow$ decrease generalization accuracy
- Large batch sizes $\Rightarrow$ 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

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?

\[ Q > 1 : \lambda^- > 0 \]
\[ Q = 1 : \lambda^- = 0 \]

\[ Q > 1 : \lambda^- > 0 \]
\[ Q = 1 : \lambda^- = 0 \]

Bulk + Spikes?  
BULK-DECAY?  
HEAVY-TAILED?

(a) Different aspect ratios

Large, well-trained, modern models approach *Heavy-tailed Self-regularization.*
Applying RMT: What phase is your model in?

Inception V3 Layer 226 \( Q \approx 1.3 \)

**Bulk-decay?**
- best MP fits
- bulk not captured

**Heavy-tailed?**
- difficult to apply MP

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 \( = 0 \)

Resembles \( Q = 1 \) fit like a soft rank collapse

Heavy-tailed, but seemingly within MP eigenvalue bounds

Variances 1.83 is quite large

Large, well-trained, modern models approach \textit{Heavy-tailed Self-regularization}. 
Applying RMT: What phase is your model in?

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

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 $Q \approx 2.048$

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

Figure 12: Distribution of power law exponents $\alpha$ 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
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 $\sim$ Tikhonov-like regularization

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

Remarkably ubiquitous

How can this be used?

Why does deep learning work?
DNN Capacity metrics: Product norms

\[ C \sim \|W_1\| \times \|W_2\| \cdots \|W_L\| \]

\[ \log C \sim \log \left[ \|W_1\| \times \|W_2\| \cdots \|W_L\| \right] \]

\[ \log C \sim \left[ \log \|W_1\| + \log \|W_2\| \cdots \log \|W_L\| \right] \]

\[ \langle \log \|W\|_F \rangle = \frac{1}{N_L} \sum_L \log \|W_L\| \]

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 $\alpha$ 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:

\[ \Pr(W_{i,j}^{\text{rand}}) \sim \frac{1}{x^{1+\mu}} \]

- Examine the norm-powerlaw relations:

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

- Argue that:

**PL-Norm Relation:** \( \alpha \log \lambda_{\text{max}} \approx \log \| W \|_F^2 \).

- 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”
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 $\alpha$, & temperature-like parameters $\tau$
- Adding noise to training data decreases $\alpha$
- Early stopping increases $\tau$

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 $\tau$-$\alpha$ plane for VSDL model.
(g) Noisifying data and adjusting knobs.

‡when analyzed via the Statistical Mechanics Theory of Generalization (SMToG)
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: $W \simeq W^{\text{rand}} + \Delta^{\text{sig}}$:
- Small/old NNs: model “noise” $W^{\text{rand}}$ with Gaussian random matrices
- Modern DNNs: model strongly-correlated “signal” $\Delta^{\text{sig}}$ 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 $\rightarrow$ Hypotheses $\rightarrow$ Build a Theory $\rightarrow$ 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
- Code: https://github.com/CalculatedContent/ImplicitSelfRegularization

Main paper (abridged):
- Traditional and Heavy-Tailed Self Regularization in Neural Network Models
- 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
- Code: https://github.com/CalculatedContent/PredictingTestAccuracies
- https://github.com/CalculatedContent/WeightWatcher
- “pip install weightwatcher”