Practice, Theory, and Theorems for Random Matrix Theory in Modern Machine Learning

Michael. W. Mahoney

ICSI and Department of Statistics University of California, Berkeley, USA

(Joint work with Charles Martin and Zhenyu Liao)

June 24, 2022

Outline

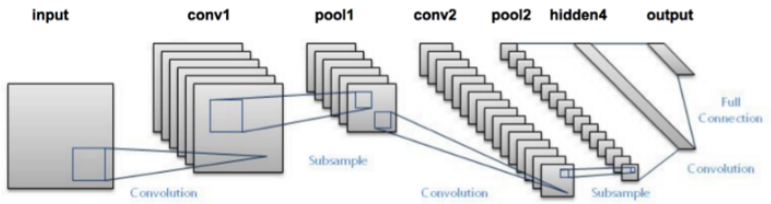
1. Practice
2. Theory
3. More practice
4. Theorems
   - Introduction
   - Sample covariance matrix for large dimensional data: from concentration to RMT
   - A random matrix perspective of the "curse of dimensionality"
   - Kernel spectral clustering for large dimensional data
   - A random matrix approach to large neural networks and random features

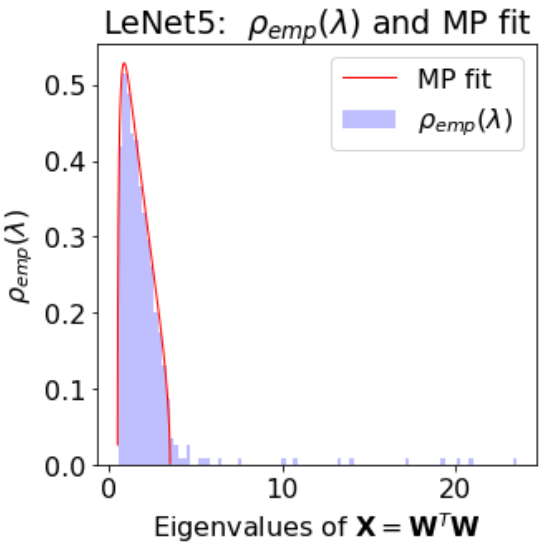
Table of Contents

Lots of DNNs analyzed: Look at nearly every publicly-available SOTA model in CV and NLP

- *Don't evaluate your method on one/two/three NNs, evaluate it on:*
  - *dozens (2017)*
  - *hundreds (2019)*
  - *thousands (2021)*
- *Don't use bad/toy models, use SOTA models.*
  - *If you do, don't be surprised if low-quality/toy models are different than high-quality/SOTA models.*
- *Don't train models, instead validate pre-trained models.*
  - *Validating models is harder than training models.*

Results: LeNet5 (an old/small NN example)





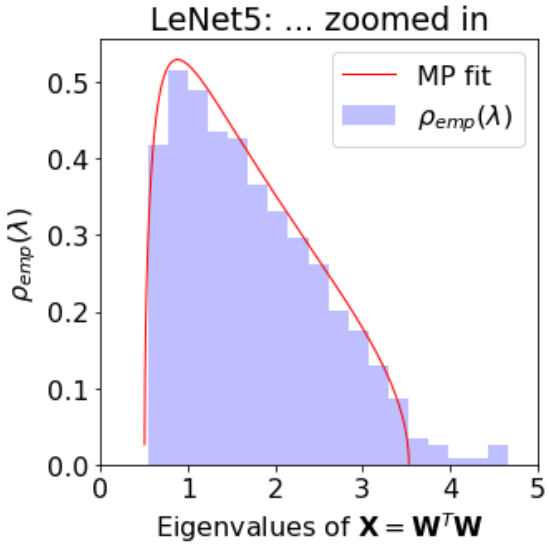
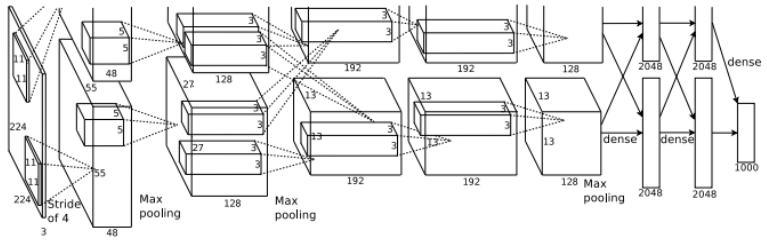
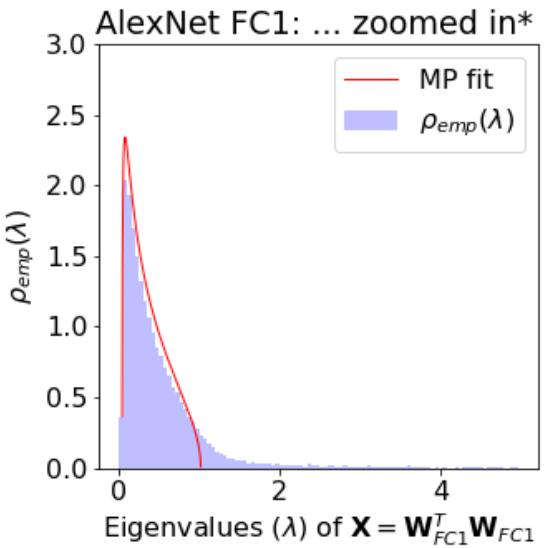


Figure: Full and zoomed-in ESD for LeNet5, Layer FC1.

Older and/or smaller and/or less well-trained models look like bulk+spike.

Results: AlexNet (a typical modern/large DNN example)





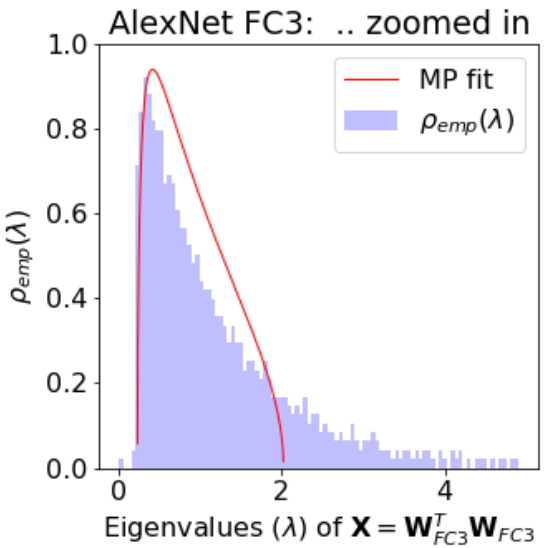


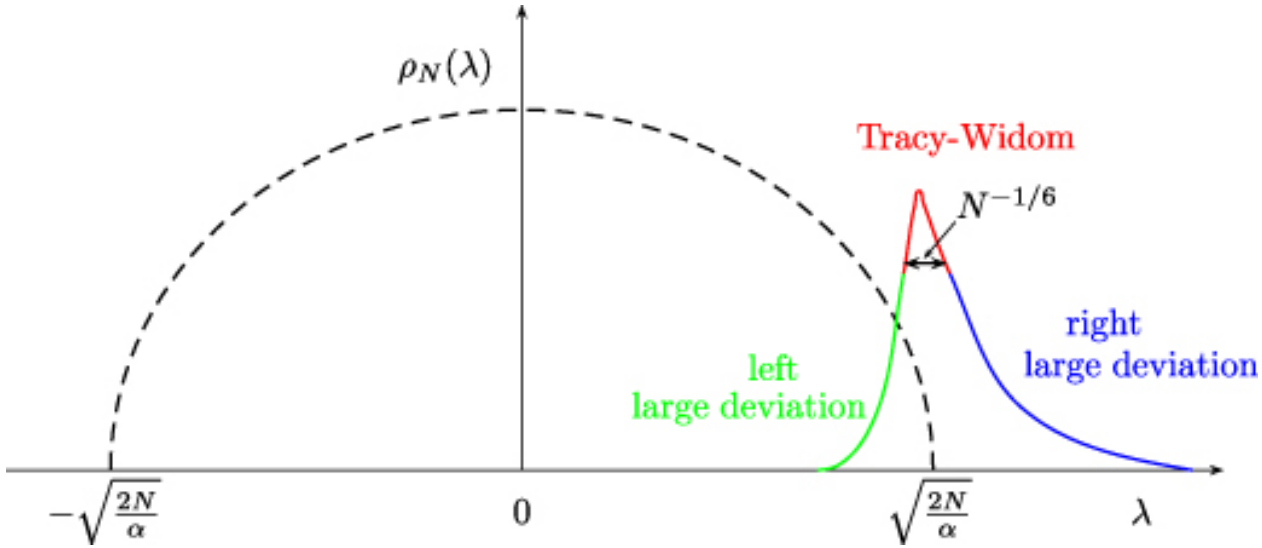
Figure: Zoomed-in ESD for Layer FC1 and FC3 of AlexNet.

Newer SOTA models have heavy-tail structure in their weight matrix correlations (i.e., not elements but eigenvalues).

Table of Contents

Random Matrix Theory 101: Wigner and Tracy-Widom

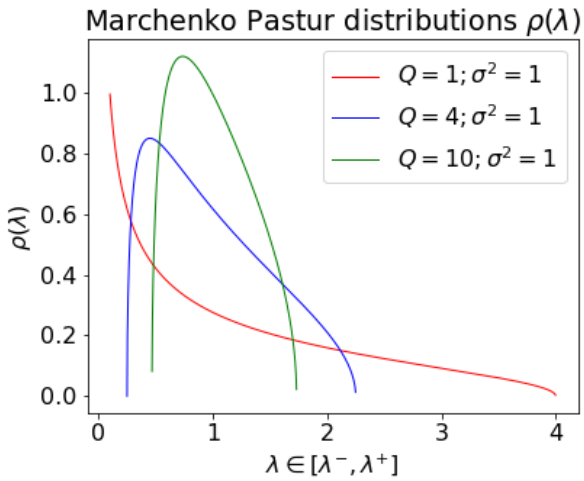
- Wigner: *global bulk statistics* approach universal semi-circular form
- Tracy-Widom: *local edge statistics* fluctuate in universal way



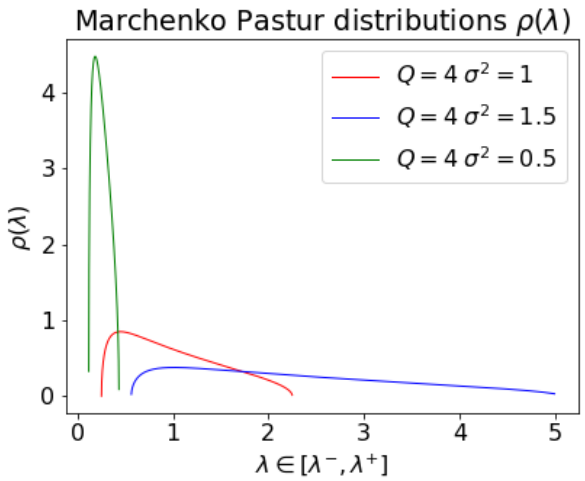
Problems with Wigner and Tracy-Widom:

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

Random Matrix Theory 102': Marchenko-Pastur



(c) Vary aspect ratios



(d) 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

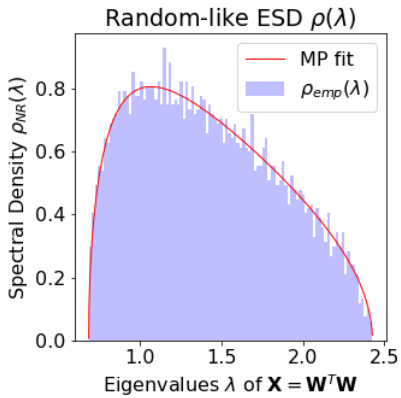
Go beyond the (relatively easy) Gaussian Universality class:

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

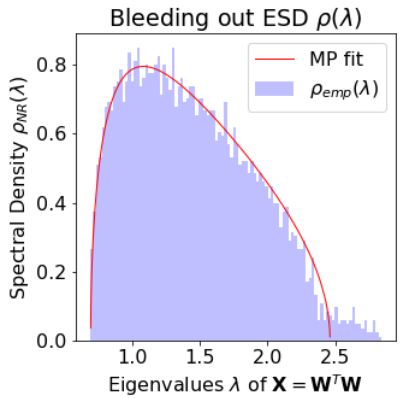
	Generative Model w/ elements from Universality class	Finite-N Global shape $\rho_N(\lambda)$	Limiting Global shape $\rho(\lambda),\ N \to \infty$	Bulk edge Local stats $\lambda \approx \lambda^+$	(far) Tail Local stats $\lambda \approx \lambda_{max}$
Basic MP	Gaussian	MP distribution	MP	TW	No tail.
Spiked-Covariance	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.

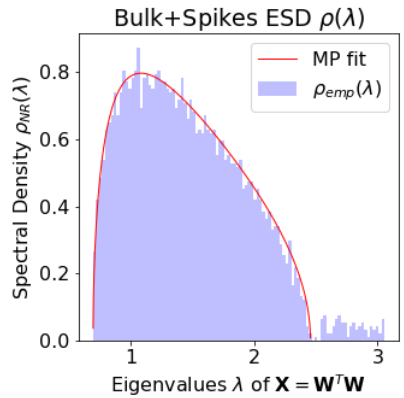
RMT-based 5+1 Phases of Training (in pictures)



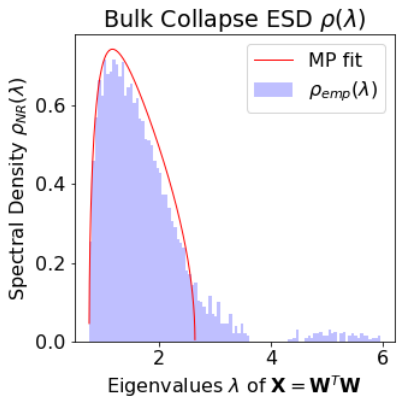
(a) RANDOM-LIKE.



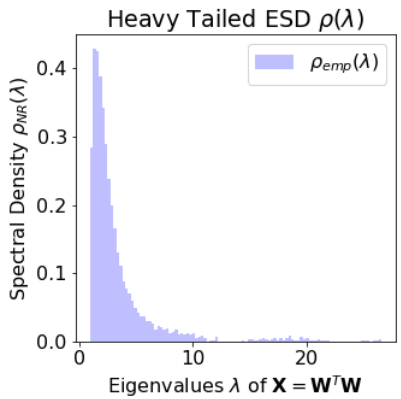
(b) BLEEDING-OUT.



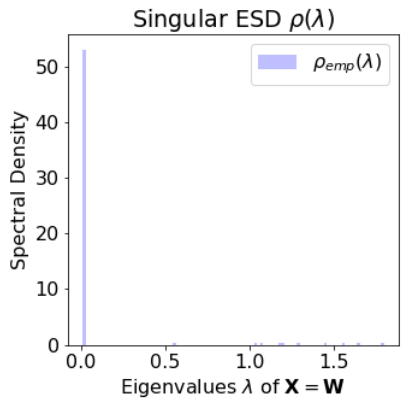
(c) BULK+SPIKES.



(d) BULK-DECAY.



(e) HEAVY-TAILED.



(f) RANK-COLLAPSE.

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

Bulk+Spikes: Small Models ∼ Tikhonov regularization

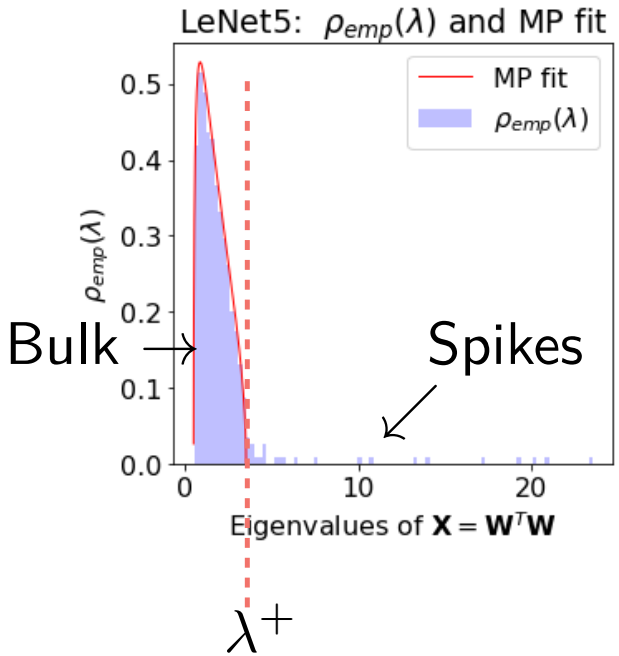
Low-rank perturbation

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

Perturbative correction

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

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



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 and can be described perturbatively with Gaussian RMT

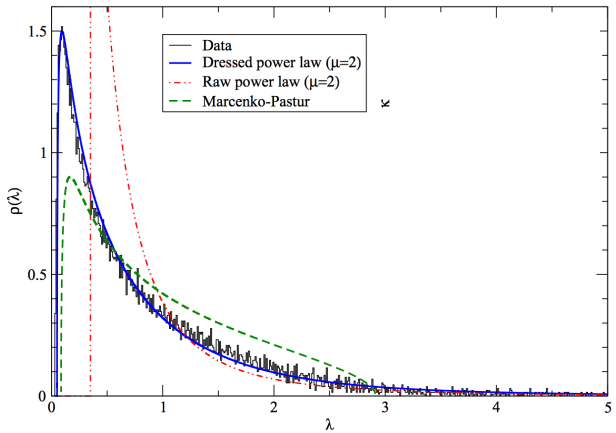
Heavy-tailed Self-regularization

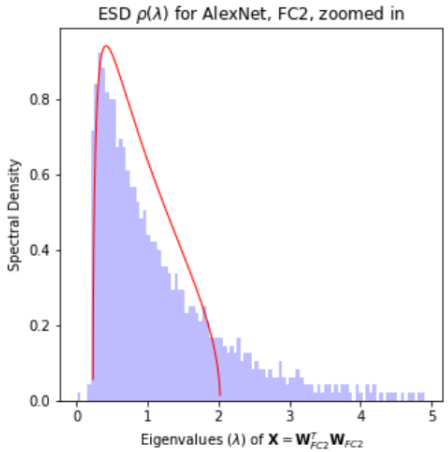
W is *strongly-correlated* and highly non-random

- We *model* strongly-correlated systems by heavy-tailed random matrices
- We *model* signal (not noise) by heavy-tailed random matrices

Then RMT/MP ESD will also have heavy tails.

- The eigenvalues are heavy-tailed; the weights are NOT.





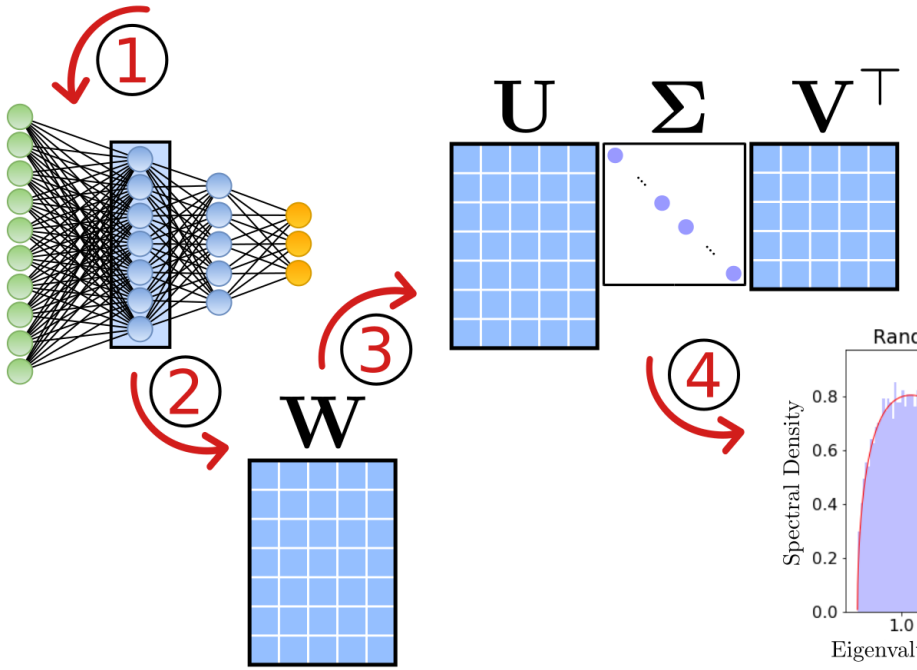
"All" larger, modern DNNs exhibit novel Heavy-tailed self-regularization

Table of Contents

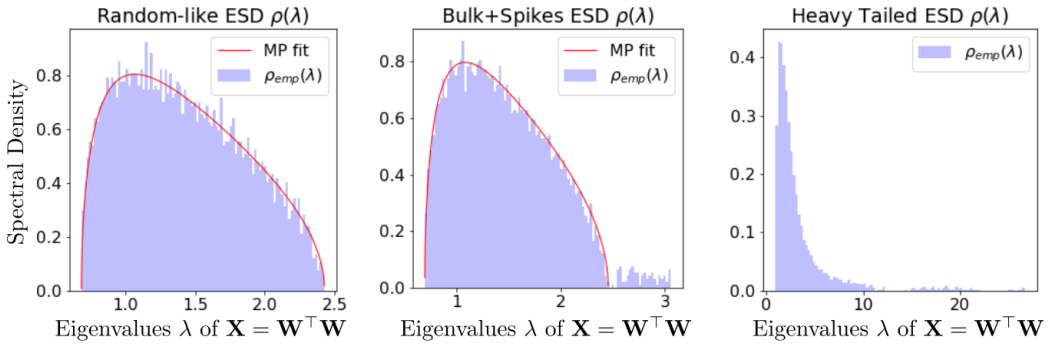
Watching weights with WeightWatcher

https://github.com/CalculatedContent/WeightWatcher

Analyzing DNN Weight matrices with WeightWatcher



1. Take a model
   - 2. Take a weight matrix
     - 3. Do Spectral analysis
       - 4. Histogram of eigenvalues



- Analyze one layer of pre-trained model
- Compare multiple layers of pre-trained model
- Monitor NN properties as you train your own model

"pip install weightwatcher"

Using the theory

Different ways one could *use* a theory.

- Perform diagnostics for model validation, to develop hypotheses, etc.*
- Make predictions about model quality, generalization, transferability, etc.*
- Did post-training modifications damage my model?*
- Will buying more data help?*
- Will training longer help?*
- Will quantizing or distilling help?*
- Construct a regularizer to do model training.**

*Ideally, by peeking at very little or no data.

**If you have lots of data, lots of GPUs, etc.

Predicting test accuracies ... lots of metrics ...

- Average log norm (a VC-like data-dependent capacity metric):

$$\langle \log \|\mathbf{W}\| \rangle = \frac{1}{N} \sum_{l,i} \log \|\mathbf{W}_{l,i}\| = \frac{1}{N} \sum_{l,i} \log(\lambda_{l,i}^{max})$$

- Average alpha (also data-dependent, from HT-SR theory):

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

- Combine the two into a weighted average (weighted to compensate for different size and scale of feature maps):

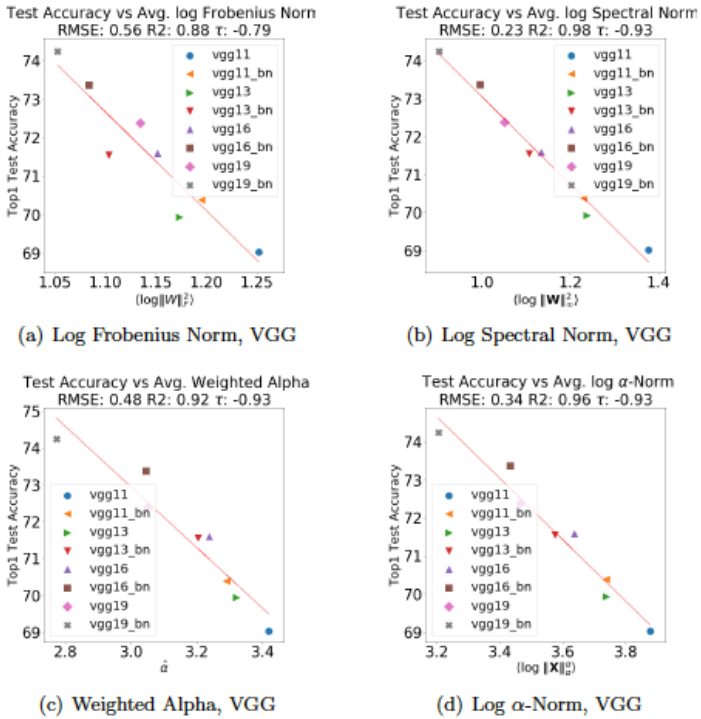
$$\hat{\alpha} = \frac{1}{N} \sum_{l,i} \log(\lambda_{l,i}^{max}) \alpha_{l,i}$$

- In a special case ($\alpha \approx 2$), for each layer:

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

"pip install weightwatcher"

(The first) large-scale study (meta-analysis) of hundreds of SOTA pretrained models [‡]



(a) Log Frobenius Norm, VGG (b) Log Spectral Norm, VGG

(c) Weighted Alpha, VGG (d) Log α-Norm, VGG

Figure 2: Comparison of Average Log Norm and Weighted Alpha quality metrics versus re test accuracy for pretrained VGG models: VGG11, VGG13, VGG16, and VGG19, with and

Different metrics on pre-trained VGG.

Series	#	Metric	$\langle \log \|\mathbf{W}\|_F^2 \rangle$	$\langle \log \|\mathbf{W}\|_\infty^2 \rangle$	$\hat{\alpha}$	$\langle \log \|\mathbf{X}\|_\alpha^\alpha \rangle$
VGG	6	RMSE	0.56	**0.23**	0.48	0.34
		R^2	0.88	**0.98**	0.92	0.96
		Kendall-τ	-0.79	**-0.93**	**-0.93**	**-0.93**
ResNet	5	RMSE	0.9	0.97	**0.61**	0.66
		R^2	0.92	0.9	**0.96**	0.9
		Kendall-τ	-1.0	-1.0	-1.0	-1.0
ResNet-1K	19	RMSE	2.4	2.8	**1.8**	1.9
		R^2	0.81	0.74	**0.89**	0.88
		Kendall-τ	-0.79	-0.79	**-0.89**	-0.88
DenseNet	4	RMSE	0.3	**0.11**	0.16	0.21
		R^2	0.93	**0.99**	0.98	0.97
		Kendall-τ	-1.0	-1.0	-1.0	-1.0

Table 1: Quality metrics (for RMSE, smaller is better; for R^2, larger is better; and for Kendall-τ rank correlation, larger magnitude is better) for reported Top1 test error for pretrained models in each architecture series. Column # refers to number of models. VGG, ResNet, and DenseNet were pretrained on ImageNet. ResNet-1K was pretrained on ImageNet-1K.

Summary statistics: VGG; ResNet; DenseNet.

	$\log \|\cdot\|_F^2$	$\log \|\cdot\|_\infty^2$	$\hat{\alpha}$	$\log \|\cdot\|_\alpha^\alpha$
RMSE (mean)	4.84	5.57	4.58	4.55
RMSE (std)	9.14	9.16	9.16	9.17
R2 (mean)	3.9	3.85	3.89	3.89
R2 (std)	9.34	9.36	9.34	9.34
Kendal-tau (mean)	3.84	3.77	3.86	3.85
Kendal-tau (std)	9.37	9.4	9.36	9.36

Table 3: Comparison of linear regression fits for different average Log Norm and Weighted Alpha metrics across 5 CV datasets, 17 architectures, covering 108 (out of over 400) different pretrained

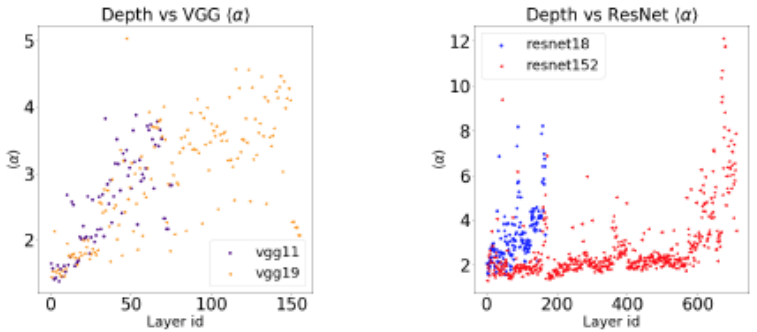
Summary statistics: hundreds of models.

Lots more plots to prove we can "predict trends . . . without access . . . "

[‡] "Predicting trends in the quality of state-of-the-art neural networks without access to training or testing data," Martin, Peng, and Mahoney, arXiv:2002.06716, *Nature Communications*, 2021.

Using a theory: on SOTA models

Analyzing pre-trained models: properties of VGG vs ResNet vs DenseNet leads to the idea of *correlation flow*.



(a) VGG (b) ResNet

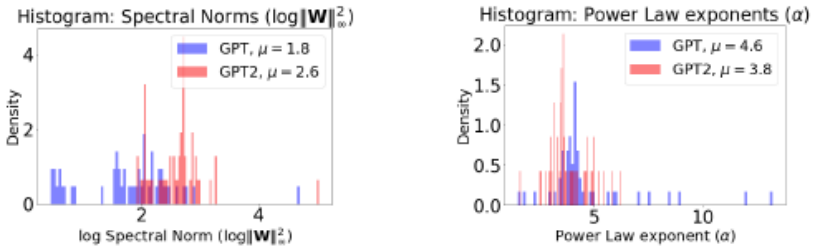
(c) DenseNet (d) ResNet (overlaid)

Figure 4: PL exponent (α) versus layer id, for the least and the most accurate models in VGG (a), ResNet (b), and DenseNet (c) series. (VGG is without BN; and note that the Y axes on

Alpha versus depth: VGG, ResNet, DenseNet.

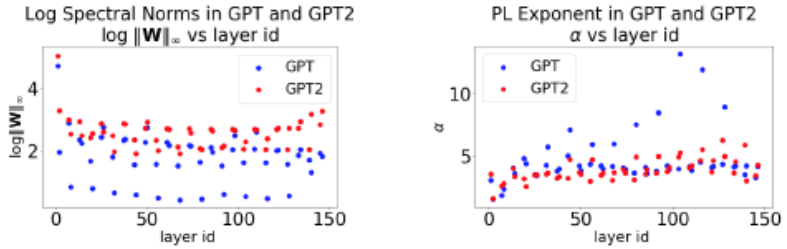
Using a theory: on SOTA models

Analyzing pre-trained models: properties of GPTx series leads to the idea of *scale collapse*.



(a) Log Spectral Norm ($\log\|\mathbf{W}\|_\infty$) (b) PL exponent ($\alpha$)

Figure 6: Histogram of PL exponents and Log Spectral Norms for weight matrices from the OpenAI GPT and GPT2-small pretrained models.



(a) Log Spectral Norm ($\log\|\mathbf{W}\|_\infty$) (b) PL exponent ($\alpha$)

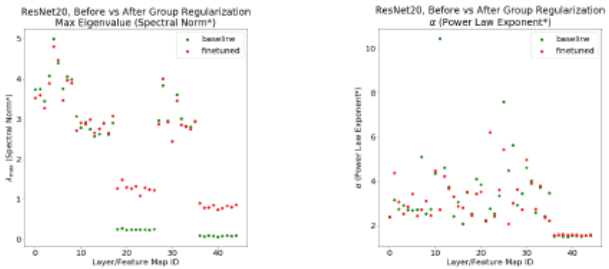
Figure 7: Log Spectral Norms (in (a)) and PL exponents (in (b)) for weight matrices from the OpenAI GPT and GPT2-small pretrained models. (Note that the quantities shown on each Y axis are different.) In the text, this is interpreted in terms of Scale Collapse and Correlation Flow.

Histogram and depth plots of $\alpha_{l,i}$ and $\lambda_{l,i}^{max}$.

Using a theory: easy to break popular SLT metrics

Easy to "break" popular SLT metrics:

- they are *not* validated counterfactually
- (but they drive the development of models)



(a) λ_{max} for ResNet20 layers (b) α for ResNet20 layers

Figure 5: ResNet20, distilled with Group Regularization, as implemented in the `distiller` (4D_regularized_5Lremoved) pretrained models. Log Spectral Norm ($\log \lambda_{max}$) and PL exponent (α) for individual layers, versus layer id, for both baseline (before distillation, green) and fine-tuned (after distillation, red) pretrained models.

Intel's distillation "broke" their models.

Series	#	$\langle \log \|\mathbf{W}\|_F \rangle$	$\langle \log \|\mathbf{W}\|_\infty \rangle$	$\hat{\alpha}$	$\langle \log \|\mathbf{X}\|_\alpha^\alpha \rangle$
GPT	49	1.64	1.72	7.01	7.28
GPT2-small	49	2.04	2.54	9.62	9.87
GPT2-medium	98	2.08	2.58	9.74	10.01
GPT2-large	146	1.85	1.99	7.67	7.94
GPT2-xl	194	1.86	1.92	7.17	7.51

Table 2: Average value for the average Log Norm and Weighted Alpha metrics for pretrained OpenAI GPT and GPT2 models. Column # refers to number of layers treated. Averages do

GPTx series: how does a model trained to "bad" data differ from one trained to "good" data?

Using a theory: leads to predictions

Based on analyzing hundreds of pre-trained SOTA models:

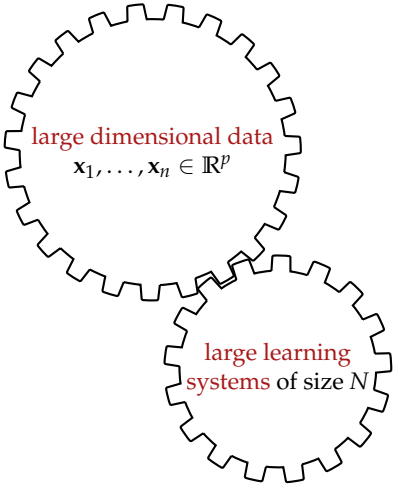
- "**Correlation flow**":
  - "Shape" of ESD of adjacent layers, as well as overlap between eigenvectors of adjecent layers, should be well-aligned.
- "**Scale collapse**":
  - "Size" of ESD of one or more layers changes dramatically, while the size of other layers changes very little, as a function of some perturbation of a model, during training (or post-training modification).
- "**Correlation traps**":
  - Spuriously large eigenvalues[§] may appear, and they may even be important for model convergence.

We can measure these quantities with Weightwatcher—so can you!

[§] Eigenvalues not due to signal in the data—we have theorems-style theory for Hessians ("Hessian Eigenspectra of More Realistic Nonlinear Models," Liao and Mahoney, `https://arxiv.org/abs/2103.01519`), but it's still open for Weights.

Table of Contents

Understanding the mechanism of large dimensional machine learning



- Big Data era: exploit large n, p, N
- counterintuitive phenomena when $n \not\gg p$, e.g., the "*curse of dimensionality*"
- complete change of understanding of many ML algorithms
- **RMT** provides the tools!

From low to high dimensional machine learning

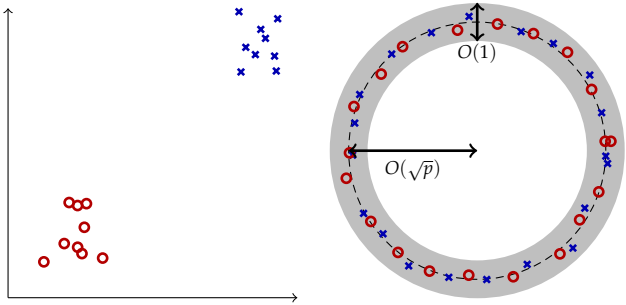


Figure: Visual representation of classification in (**left**) small and (**right**) large dimensions.

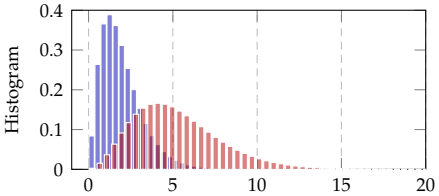
- **low dimension**: data vectors $\mathbf{x}_i \in \mathbb{R}^p, p = 2, 3$, gathered in different "groups" can be classified using distance-based approach
- **high dimension**:
  - (i) **easy** or **trivial** scenario where low dimensional intuition holds and a pairwise distance-based classification approach via, e.g., Johnson–Lindenstrauss lemma, is efficient;
  - (ii) **hard** or **non-trivial** scenario where such intuition collapses: data vectors at approximately the **same** Euclidean distance, **regardless** their arising from *same or different* classes.

Non-trivial high dimensional classification beyond the JL regime

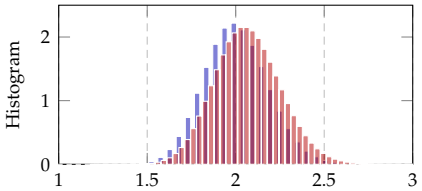
In the high dimensional regime where data dimension p and sample size n both large, a **dual** phenomenon:

(i) data points not pairwise classifiable: Euclidean distance between any two data points $\mathbf{x}_i \in \mathcal{C}_a$ and $\mathbf{x}_j \in \mathcal{C}_b$ approximately constant $\approx \tau = O(1)$ independent of their classes $\mathcal{C}_a, \mathcal{C}_b$: $\|\mathbf{x}_i - \mathbf{x}_j\|^2/p = \tau + o(1)$ as $n, p \to \infty$ and data pairs *neither close nor far* from each other;

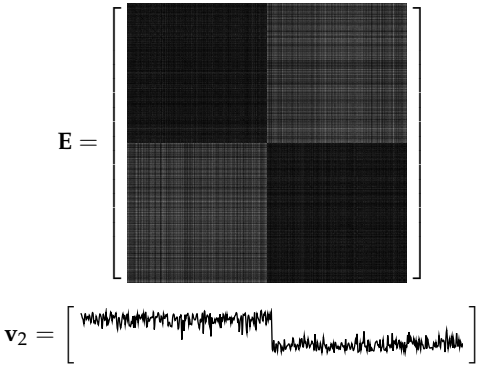
(ii) classification remains possible by exploiting the spectral information of large Euclidean distance **matrix** $\mathbf{E} = \{\|\mathbf{x}_i - \mathbf{x}_j\|^2/p\}_{i,j=1}^n$, thanks to a collective behavior of all data belonging to same (and large) classes.



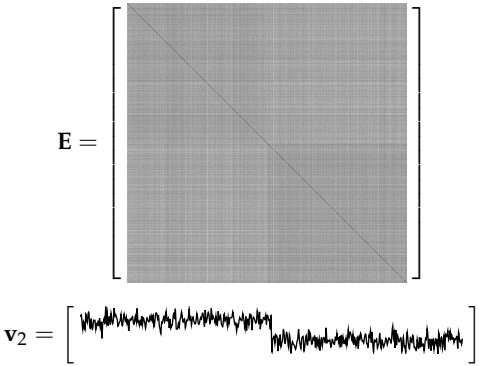
(a) $p = 5$



(b) $p = 250$



(a) $p = 5$



(b) $p = 250$

Figure: Euclidean distance matrices $\mathbf{E}$, the histogram of the entries of $\mathbf{E}$, and the second top eigenvectors $\mathbf{v}_2$, for small (**left**, $p = 5$) and large (**right**, $p = 250$) dimensional data $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{p \times n}$ with $\mathbf{x}_1, \ldots, \mathbf{x}_{n/2} \in \mathcal{C}_1$ and $\mathbf{x}_{n/2+1}, \ldots, \mathbf{x}_n \in \mathcal{C}_2$ for $n = 5\,000$ and different values of p.

Sample covariance matrix in the large n, p regime

- For $\mathbf{x}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$, estimate population covariance $\mathbf{C} \in \mathbb{R}^{p \times p}$ from n data samples $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{p \times n}$.

- Maximum likelihood sample covariance matrix with entry-wise convergence

$$\hat{\mathbf{C}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^{\mathsf{T}} = \frac{1}{n} \mathbf{X}\mathbf{X}^{\mathsf{T}} \in \mathbb{R}^{p \times p}, \quad [\hat{\mathbf{C}}]_{ij} \to [\mathbf{C}]_{ij}$$

  almost surely as $n \to \infty$: optimal for $n \gg p$ (or, for p "small").

- In the regime $n \sim p$, conventional wisdom breaks down:
  for $\mathbf{C} = \mathbf{I}_p$ with $n < p$, $\hat{\mathbf{C}}$ has at least $p - n$ zero eigenvalues.

$$\boxed{\|\hat{\mathbf{C}} - \mathbf{C}\| \not\to 0, \quad n, p \to \infty}$$

  $\Rightarrow$ eigenvalue mismatch and not consistent! $\Rightarrow$ **matrix norms not equivalent in large dimensions!**
- due to $\|\mathbf{A}\|_\infty \leq \|\mathbf{A}\| \leq p\|\mathbf{A}\|_\infty$ for $\mathbf{A} \in \mathbb{R}^{p \times p}$ and $\|\mathbf{A}\|_\infty \equiv \max_{ij} |\mathbf{A}_{ij}|$.

Quantitative spectral characterization of sample covariance

Theorem (Concentration of sample covariance, [Ver18, Theorem 4.6.1])

Let $\mathbf{X} \in \mathbb{R}^{p\times n}$ be a random matrix with i.i.d. sub-gaussian columns $\mathbf{x}_i \in \mathbb{R}^p$ such that $\mathbb{E}[\mathbf{x}_i] = \mathbf{0}$ and $\mathbb{E}[\mathbf{x}_i\mathbf{x}_i^\mathsf{T}] = \mathbf{I}_p$, one has, with probability at least $1 - 2\exp(-t^2)$ for any $t \geq 0$ that

$$\|\hat{\mathbf{C}} - \mathbf{I}_p\| \leq C_1 \max(\delta, \delta^2), \quad \delta = C_2(\sqrt{p/n} + t/\sqrt{n}) \tag{1}$$

for some constants $C_1, C_2 > 0$ independent of n, p.

- non-asymptotic and high probability characterization
- however, not precise in the $p \sim n$ regime, since $\delta = O(\sqrt{p/n}) = O(1)$

Theorem (Marčenko-Pastur law, [MP67])

Under the same setting of Theorem 1, as $n, p \to \infty$ with $p/n \to c \in (0, \infty)$, with probability one, the empirical spectral measure $\mu_{\hat{\mathbf{C}}} \equiv \frac{1}{p}\sum_{i=1}^p \delta_{\lambda_i(\hat{\mathbf{C}})}$ of $\hat{\mathbf{C}} \equiv \frac{1}{n}\mathbf{X}\mathbf{X}^\mathsf{T}$ converges weakly to a probability measure μ given explicitly by

$$\mu(dx) = (1 - c^{-1})^+\delta_0(x) + \frac{1}{2\pi cx}\sqrt{(x - E_-)^+(E_+ - x)^+}\,dx \tag{2}$$

where $E_\pm = (1 \pm \sqrt{c})^2$ and $(x)^+ = \max(0, x)$, and is known as the Marčenko-Pastur law.

Two ways of spectral characterization of sample covariance

- matrix concentration-type characterization

$$\|\hat{\mathbf{C}} - \mathbf{I}_p\| \leq C_1 \max(\delta, \delta^2), \quad \delta = C_2(\sqrt{p/n} + t/\sqrt{n})$$

  $\Rightarrow$ non-asymptotic characterization of **small dimensional** intuition: how $\hat{\mathbf{C}}$ concentrates around $\mathbf{I}_p$;
- random matrix-type characterization of precise eigenvalue distribution

$$\mu(dx) = (1 - c^{-1})^+ \delta(x) + \frac{1}{2\pi c x} \sqrt{(x - E_-)^+ (E_+ - x)^+} dx$$

  $\Rightarrow$ asymptotic characterization (as $n, p \to \infty$) of **large dimensional** intuition: how $\hat{\mathbf{C}}$ differs from $\mathbf{I}_p$!

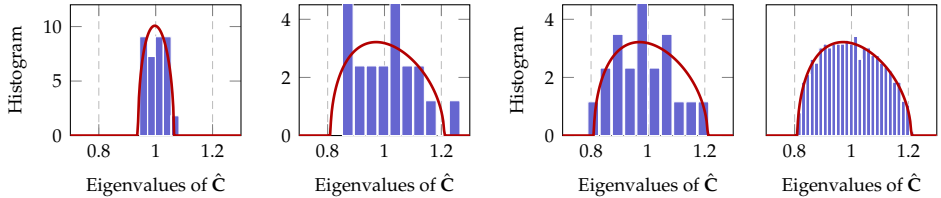


Figure: Histogram of the eigenvalues of $\hat{\mathbf{C}}$ (**blue**) versus the Marčenko-Pastur law (**red**), for $\mathbf{X}$ having standard Gaussian entries in different settings: (**left**: small versus large dimensional intuition) $p = 20, n = 1\,000p$ versus $p = 20, n = 100p$; and (**right**: non-asymptotic versus asymptotic MP law) $p = 20, n = 100p$ versus $p = 500, n = 100p$.

When is one in the random matrix regime? Almost always!

What about $n = 100p$? For $\mathbf{C} = \mathbf{I}_p$, as $n, p \to \infty$ with $p/n \to c \in (0, \infty)$: the Marčenko–Pastur law

$$\mu(dx) = (1 - c^{-1})^+ \delta(x) + \frac{1}{2\pi c x} \sqrt{(x - E_-)^+ (E_+ - x)^+} dx$$

where $E_- = (1 - \sqrt{c})^2$, $E_+ = (1 + \sqrt{c})^2$ and $(x)^+ \equiv \max(x, 0)$. Close match!

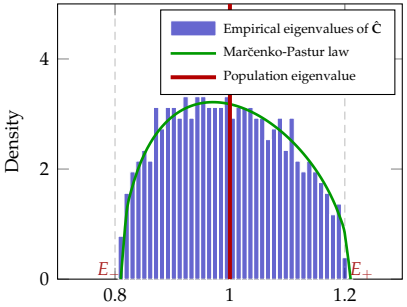


Figure: Eigenvalue distribution of $\hat{\mathbf{C}}$ versus Marčenko-Pastur law, $p = 500$, $n = 50\,000$.

- eigenvalues span on $[E_- = (1 - \sqrt{\mathbf{c}})^2, E_+ = (1 + \sqrt{\mathbf{c}})^2]$.
- for $\mathbf{n = 100p}$, on a range of $\pm\mathbf{2}\sqrt{\mathbf{c}} = \pm\mathbf{0.2}$ around the population eigenvalue **1**.

Beyond eigenvalue distribution: a modern RMT approach via the resolvent

This **change-of-intuition** leads to very different behavior for small- versus large-dimensional ML:

- **linear** models: low-rank approximation, spectral classification/clustering, and linear least squares regression in high dimensions different from their small dimensional counterparts
- as well as more involved **nonlinear** models: kernel spectral clustering, nonlinear neural nets, etc.

Technical challenges:

- classical RMT focuses on eigenvalue distribution
- ML applications need **eigenvectors** and more complex matrix **functionals**!

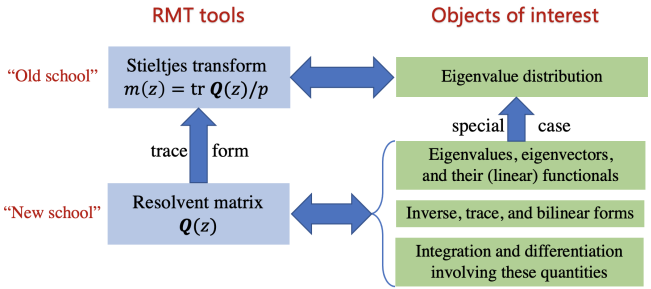


Figure: Different objects of interest and their corresponding technical tools for "old" and "new school" RMT.

"Curse of dimensionality": loss of relevance of Euclidean distance

- Binary Gaussian mixture classification $\mathbf{x} \in \mathbb{R}^p$:

$$\mathcal{C}_1 : \mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_1, \mathbf{C}_1), \text{ versus } \mathcal{C}_2 : \mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_2, \mathbf{C}_2);$$

- Neyman-Pearson test: classification is possible only when [CLM18]

$$\boxed{\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\| \geq C_{\boldsymbol{\mu}}, \text{ or } \|\mathbf{C}_1 - \mathbf{C}_2\| \geq C_{\mathbf{C}} \cdot p^{-1/2}}$$

  for some constants $C_{\boldsymbol{\mu}}, C_{\mathbf{C}} > 0$.

- In this non-trivial setting, for $\mathbf{x}_i \in \mathcal{C}_a, \mathbf{x}_j \in \mathcal{C}_b$:

$$\boxed{\max_{1 \leq i \neq j \leq n} \left\{ \frac{1}{p} \|\mathbf{x}_i - \mathbf{x}_j\|^2 - \frac{2}{p} \operatorname{tr} \mathbf{C}^\circ \right\} \xrightarrow{a.s.} 0}$$

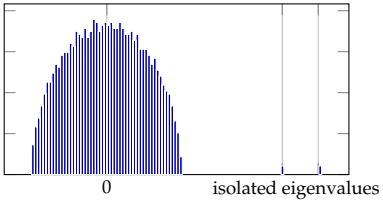
  as $n, p \to \infty$ (i.e., $n \sim p$), for $\mathbf{C}^\circ \equiv \frac{1}{2}(\mathbf{C}_1 + \mathbf{C}_2)$, regardless of the classes $\mathcal{C}_a, \mathcal{C}_b$! (In fact even for $n = p^m$.)

$\Rightarrow$ Direct consequence to various distance-based machine learning methods (e.g., kernel spectral clustering)!

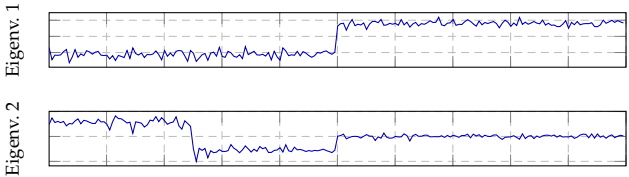
[1] Romain Couillet, Zhenyu Liao, and Xiaoyi Mai. "Classification asymptotics in the random matrix regime". In: *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE. 2018, pp. 1875–1879

Reminder on kernel spectral clustering

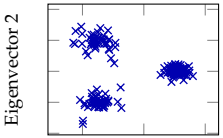
Two-step classification of n data points based on distance kernel matrix $\mathbf{K} \equiv \{f(\|\mathbf{x}_i - \mathbf{x}_j\|^2/p)\}_{i,j=1}^n$:



Reminder on kernel spectral clustering



$\Downarrow$ K**-dimensional representation** $\Downarrow$



$\Downarrow$

EM or k-means clustering.

(Three classes/clusters in this example.)

Visualization of kernel matrices for large dimensional Gaussian data

Objective: "cluster" Gaussian data $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbf{R}^p$ into $\mathcal{C}_1$ or $\mathcal{C}_2$.

Consider Gaussian kernel matrix $\mathbf{K}_{ij} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2p)$ and the second top eigenvectors $\mathbf{v}_2$ for small (**left**) and large (**right**) dimensional data.

(a) $p = 5, n = 500$ (b) $p = 250, n = 500$

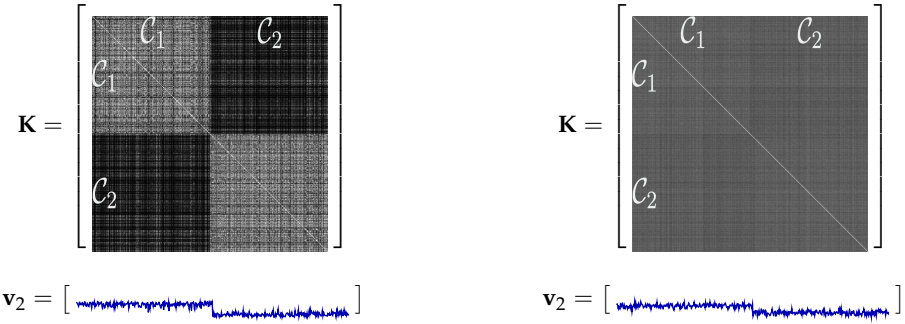
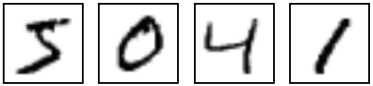


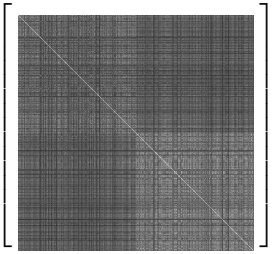
Figure: Kernel matrices $\mathbf{K}$ and the second top eigenvectors $\mathbf{v}_2$ for small (left, $p = 5, n = 500$) and large (right, $p = 250, n = 500$) dimensional data.

Kernel matrices for large dimensional real-world data

(a) MNIST

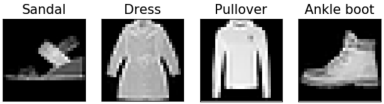


$\mathbf{K} =$

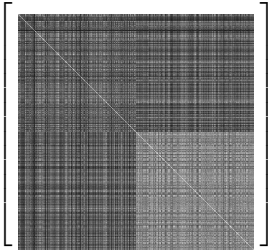


$\mathbf{v}_2 =$

(b) Fashion-MNIST



$\mathbf{K} =$



$\mathbf{v}_2 =$

A spectral viewpoint of large kernel matrices in large dimensions

- "local" linearization of *nonlinear* kernel matrices in large dimensions, e.g., Gaussian kernel matrix $\mathbf{K}_{ij} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2p)$ with $\mathbf{C}_1 = \mathbf{C}_2 = \mathbf{I}_p$ (e.g., $\mathcal{C}_1 : \mathbf{x}_i = \boldsymbol{\mu}_1 + \mathbf{z}_i$ versus $\mathcal{C}_2 : \mathbf{x}_j = \boldsymbol{\mu}_2 + \mathbf{z}_j$) so that

$$\|\mathbf{x}_i - \mathbf{x}_j\|^2/p \xrightarrow{a.s.} 2, \text{ and } \mathbf{K} = \exp\left(-\frac{2}{2}\right)\left(\mathbf{1}_n\mathbf{1}_n^\mathsf{T} + \frac{1}{p}\mathbf{Z}^\mathsf{T}\mathbf{Z}\right) + g(\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|)\frac{1}{p}\mathbf{j}\mathbf{j}^\mathsf{T} + * + o_{\|\cdot\|}(1)$$

  with Gaussian matrix $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_n] \in \mathbb{R}^{p\times n}$ and $\mathbf{j} = [\mathbf{1}_{n/2}; -\mathbf{1}_{n/2}]$, the class-information vector
- accumulated effect of small "hidden" statistical information ($\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|$ in this case)

Therefore

- **entry-wise**:

$$\mathbf{K}_{ij} = \exp(-1)\Big(1 + \underbrace{\frac{1}{p}\mathbf{z}_i^\mathsf{T}\mathbf{z}_j}_{O(p^{-1/2})}\Big) \pm \underbrace{\frac{1}{p}g(\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|)}_{O(p^{-1})} + *, \text{ so that } \frac{1}{p}g(\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|) \ll \frac{1}{p}\mathbf{z}_i^\mathsf{T}\mathbf{z}_j,$$

- **spectrum-wise**: (i) $\|\mathbf{K} - \exp(-1)\mathbf{1}_n\mathbf{1}_n^\mathsf{T}\| \not\to 0$; (ii) $\|\frac{1}{p}\mathbf{Z}^\mathsf{T}\mathbf{Z}\| = O(1)$ and $\|g(\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|)\frac{1}{p}\mathbf{j}\mathbf{j}^\mathsf{T}\| = O(1)$!
- **Same** phenomenon as the sample covariance example: $[\hat{\mathbf{C}} - \mathbf{C}]_{ij} \to 0 \not\Rightarrow \|\hat{\mathbf{C}} - \mathbf{C}\| \to 0$!

$\Rightarrow$ With **modern RMT**, we understand kernel spectral clustering (eigenvectors!) for large dimensional data!

Reminder on random features and neural networks

- kernel matrices $\mathbf{K} \in \mathbb{R}^{n \times n}$ from pairwise comparison of n data points: expansive for n large
- **idea**: find easy-to-compute $\hat{\mathbf{K}}$ to approximate $\mathbf{K}$, e.g., $\|\hat{\mathbf{K}} - \mathbf{K}\|$ is small
- **example**: random Fourier feature [RR08][3] $\mathbf{\Sigma}^\mathsf{T} = [\cos(\mathbf{W}\mathbf{X})^\mathsf{T},\ \sin(\mathbf{W}\mathbf{X})^\mathsf{T}] \in \mathbb{R}^{2N \times n}$ of data $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{p \times n}$ with standard Gaussian $\mathbf{W} \in \mathbb{R}^{N \times p}$, i.e., $\mathbf{W}_{ij} \sim \mathcal{N}(0,1)$
- approximates Gaussian kernel $\exp(\|\mathbf{x}_i - \mathbf{x}_j\|^2/2)$: entry-wise convergence of RFF Gram $\frac{1}{N}[\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma}]_{ij} \to [\mathbf{K}_{\text{Gauss}}]_{ij}$ Gaussian kernel matrix as number of features $N \to \infty$
- **proof**: (strong) law of large numbers:

$$\frac{1}{N}[\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma}]_{ij} = \frac{1}{N}\sum_{k=1}^{N} \cos(\mathbf{x}_i^\mathsf{T}\mathbf{w}_k)\cos(\mathbf{w}_k^\mathsf{T}\mathbf{x}_j) + \sin(\mathbf{x}_i^\mathsf{T}\mathbf{w}_k)\sin(\mathbf{w}_k^\mathsf{T}\mathbf{x}_j)$$
$$\to \mathbb{E}_{\mathbf{w}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_p)}[\cos(\mathbf{x}_i^\mathsf{T}\mathbf{w})\cos(\mathbf{w}^\mathsf{T}\mathbf{x}_j) + \sin(\mathbf{x}_i^\mathsf{T}\mathbf{w})\sin(\mathbf{w}^\mathsf{T}\mathbf{x}_j)] = [\mathbf{K}_{\cos} + \mathbf{K}_{\sin}]_{ij} = [\mathbf{K}_{\text{Gauss}}]_{ij}$$

for $\mathbf{K}_{\cos} = e^{-\frac{1}{2}(\|\mathbf{x}_i\|^2 + \|\mathbf{x}_j\|^2)}\cosh(\mathbf{x}_i^\mathsf{T}\mathbf{x}_j)$ and $\mathbf{K}_{\sin} = e^{-\frac{1}{2}(\|\mathbf{x}_i\|^2 + \|\mathbf{x}_j\|^2)}\sinh(\mathbf{x}_i^\mathsf{T}\mathbf{x}_j)$.

[3] Ali Rahimi and Benjamin Recht. "Random Features for Large-Scale Kernel Machines". In: *Advances in Neural Information Processing Systems*. Vol. 20. NIPS'08. Curran Associates, Inc., 2008, pp. 1177–1184

Random features-based ridge regression and neural networks

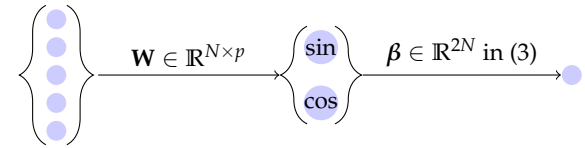


Figure: Illustration of random Fourier features regression model.

- RFF ridge regressor $\boldsymbol{\beta} \in \mathbb{R}^{2N}$ given by, for regularization penalty $\gamma \geq 0$,

$$\boldsymbol{\beta} \equiv \frac{1}{n}\boldsymbol{\Sigma}\left(\frac{1}{n}\boldsymbol{\Sigma}^{\mathsf{T}}\boldsymbol{\Sigma} + \gamma \mathbf{I}_n\right)^{-1}\mathbf{y} \cdot 1_{2N>n} + \left(\tfrac{1}{n}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{\mathsf{T}} + \gamma \mathbf{I}_{2N}\right)^{-1}\tfrac{1}{n}\boldsymbol{\Sigma}\,\mathbf{y} \cdot 1_{2N<n}. \tag{3}$$

- **Performance**: training and test Mean Squared Error (MSE): $E_{\text{train}} = \frac{1}{n}\|\mathbf{y} - \boldsymbol{\Sigma}_{\mathbf{X}}^{\mathsf{T}}\boldsymbol{\beta}\|^2$ and $E_{\text{test}} = \frac{1}{\hat{n}}\|\hat{\mathbf{y}} - \boldsymbol{\Sigma}_{\hat{\mathbf{X}}}^{\mathsf{T}}\boldsymbol{\beta}\|^2$, with $\boldsymbol{\Sigma}_{\hat{\mathbf{X}}}^{\mathsf{T}} \in \mathbb{R}^{\hat{n}\times 2N}$ RFFs of a test set $(\hat{\mathbf{X}}, \hat{\mathbf{y}})$ of size $\hat{n}$.
- single-hidden-layer neural network with $\cos + \sin$ activations, connected to neural tangent kernel (NTK)[3]

[3] Arthur Jacot, Franck Gabriel, and Clément Hongler. "Neural Tangent Kernel: Convergence and Generalization in Neural Networks". In: *Advances in Neural Information Processing Systems*. Vol. 31. NIPS'18. Curran Associates, Inc., 2018, pp. 8571–8580

Random Fourier features approximate Gaussian kernel, but in which sense?

- [RR08]: entry-wise convergence of RFF Gram $\frac{1}{N}[\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma}]_{ij} \to [\mathbf{K}_{\text{Gauss}}]_{ij}$ Gaussian kernel matrix as $N \to \infty$
- again, **not true** in spectral norm sense, i.e., $\|\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma}/N - \mathbf{K}_{\text{Gauss}}\| \not\to 0$ unless $N \gg n$
  - e.g., $\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma} \in \mathbb{R}^{n\times n}$ of rank at most N if $N \leq n$, while $\mathbf{K}_{\text{Gauss}}$ of rank n (for distinct $\mathbf{x}_i$)
  - significant impact on various RFF-based algorithms

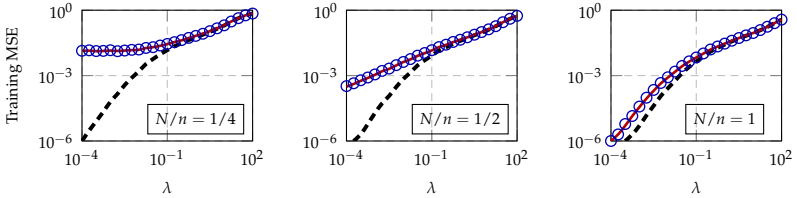


Figure: Training MSEs of RFF ridge regression on MNIST data (class 3 versus 7) as a function of regression penalty λ.

- effective kernel can be derived with RMT in the large n, p, N regime
- provides precise training and test performances of RFF for any ratio N/n, more practical and more flexible, recover Gaussian kernel result with $N/n \to \infty$
- data-dependent theory with **no** strong assumption on data

Sharp analysis of RFF ridge regression performance via RMT

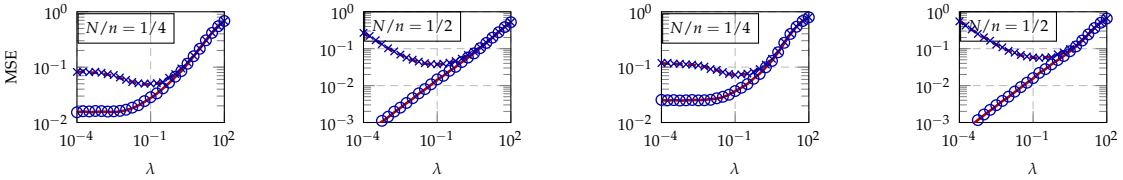


Figure: MSEs of RFF ridge regression on Fashion- (**left two**) and Kannada-MNIST (**right two**).

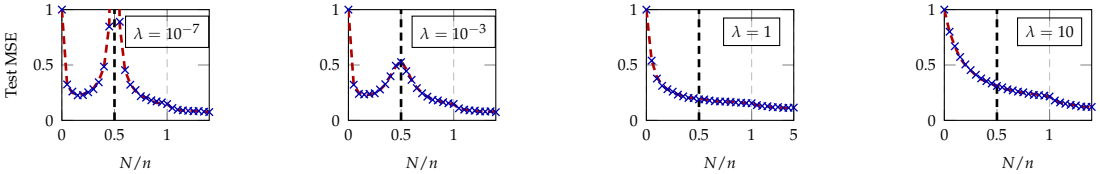
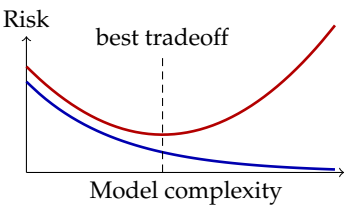
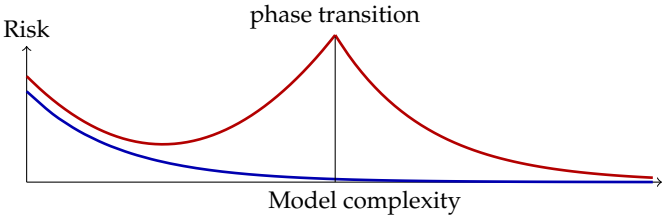


Figure: Test MSEs of RFF regression as a function of the ratio N/n, on MNIST data set.

"Recap" for double descent phenomenon for over-parameterized models



(a) Classical U-shaped risk



(b) Modern "double descent"risk

Figure: Comparison between training risk (blue) and true/test risk (red).

- empirically observed for various large-scale machine learning models, e.g., RF-based methods, decision trees, ensemble methods, and deep NNs
- **proved** here for RFF on real-world data!
- phase transition from under- to over-param of resolvent $(\mathbf{\Sigma}^{\mathsf{T}}\mathbf{\Sigma} + \lambda \mathbf{I}_n)^{-1}$ in the ridgeless $\lambda \to 0$ limit

[4] Mikhail Belkin et al. "Reconciling modern machine-learning practice and the classical bias–variance trade-off". In: *Proceedings of the National Academy of Sciences* 116.32 (2019), pp. 15849–15854

[5] Trevor Hastie et al. "Surprises in High-Dimensional Ridgeless Least Squares Interpolation". In: *arXiv* (2019). eprint: 1903.08560

Take-away messages and references

Take-away messages:

- RF methods: classical statistical learning theory provides performance guarantee for $N \gg n, p$
- here we derive (limiting) kernel in the more practical **large** n, p, N **regime**
- fast tuning of regularization parameter λ
- double descent theory for novel understanding of **over-parameterized** neural networks

References:

- **Zhenyu Liao**, Romain Couillet, and Michael W Mahoney. "A random matrix analysis of random Fourier features: beyond the Gaussian kernel, a precise phase transition, and the corresponding double descent". In: *Advances in Neural Information Processing Systems (NeurIPS)*. vol. 33. Curran Associates, Inc., 2020, pp. 13939–13950
- Cosme Louart, **Zhenyu Liao**, and Romain Couillet. "A random matrix approach to neural networks". In: *Annals of Applied Probability* 28.2 (2018), pp. 1190–1248
- Song Mei and Andrea Montanari. "The Generalization Error of Random Features Regression: Precise Asymptotics and the Double Descent Curve". In: *Communications on Pure and Applied Mathematics* (2021)
- **Zhenyu Liao** and Michael W. Mahoney. "Hessian Eigenspectra of More Realistic Nonlinear Models". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2021

Random matrix theory (RMT) for machine learning:

- **change of intuition** from small to large dimensional learning paradigm!
- **better understanding** of existing methods: why they work if they do, and what the issue is if they do not
- **improved novel methods** with performance guarantee!

Thank you! Q & A?