Random Matrix Methods for Machine Learning: "Lossless" Compression of Large and Deep Neural Networks

@ School of Physical & Mathematical Sciences, NTU

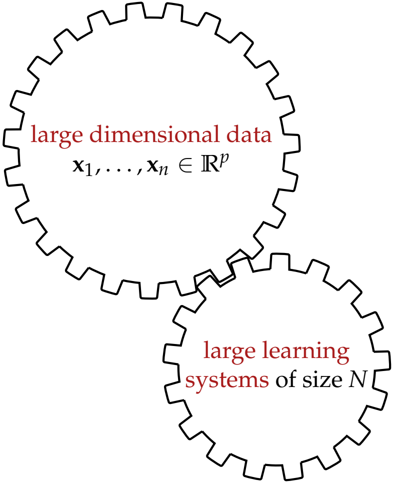
Zhenyu Liao

School of Electronic Information and Communications
Huazhong University of Science and Technology (HUST)

January 18, 2023

Outline

Motivation: understanding the mechanism of large dimensional machine learning



- **Big Data** era: exploit large n, p, N
- ImageNet dataset (`http://www.image-net.org/`): in average $p = 0.2$ million pixels of in total $n = 14$ million high-resolution images
- counterintuitive phenomena different from classical asymptotic statistics ($p \ll n$), e.g., the "curse of dimensionality"
- complete change of understanding of many algorithms
- **Random Matrix Theory** (**RMT**) provides the tools!

"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]

$$\boxed{\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\| \geq C_1, \text{ or } \|\mathbf{C}_1 - \mathbf{C}_2\| \geq C_2 \cdot p^{-1/2}}$$

for some constants $C_1, C_2 > 0$.

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

$$\boxed{\max_{1 \leq i \neq j \leq n} \left\{ \frac{1}{p} \mathbf{x}_i^\mathsf{T} \mathbf{x}_j \right\} \to 0 \quad \text{and} \quad \max_{1 \leq i \neq j \leq n} \left\{ \frac{1}{p} \|\mathbf{x}_i - \mathbf{x}_j\|^2 - \tau \right\} \to 0}$$

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

[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

Loss of relevance of Euclidean distance in large dimensions: visual representation

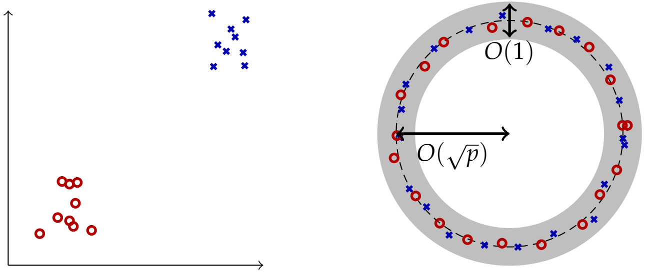
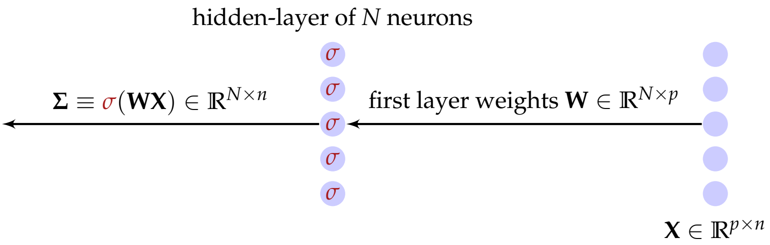


Figure: Visual representation of classification in (**left**) small and (**right**) large dimensions.

$\Rightarrow$ Direct consequence to various angle- and/or distance-based machine learning methods!

System model: a random single-hidden-layer neural network



- **Key object**: $\frac{1}{N}\boldsymbol{\Sigma}^\mathsf{T}\boldsymbol{\Sigma}$, correlation in the feature space, for random first-layer weights, e.g., $\mathbf{W}_{ij} \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$
- $\frac{1}{N}\boldsymbol{\Sigma}^\mathsf{T}\boldsymbol{\Sigma} = \frac{1}{N}\sum_{i=1}^N \sigma(\mathbf{X}^\mathsf{T}\mathbf{w}_i)\sigma(\mathbf{w}_i^\mathsf{T}\mathbf{X})$ for independent $\mathbf{w}_i$ such that $\mathbb{E}[\mathbf{w}_i] = \mathbf{0}$ and $\mathbb{E}[\mathbf{w}_i\mathbf{w}_i^\mathsf{T}] = \mathbf{I}_p$.
- **Performance guarantee**: e.g., in the infinite-neuron limit ($N \to \infty$), depends on the expected kernel matrix (and [LLC18] beyond the $N \gg \max(n,p)$ setting)

$$\boxed{\frac{1}{N}\boldsymbol{\Sigma}^\mathsf{T}\boldsymbol{\Sigma} \to \mathbf{K}(\mathbf{X}) \equiv \mathbb{E}_\mathbf{w}[\sigma(\mathbf{X}^\mathsf{T}\mathbf{w})\sigma(\mathbf{w}^\mathsf{T}\mathbf{X})] \in \mathbb{R}^{n \times n}}$$

Question: compression by carefully choosing **weights W** and/or **activation?** $\sigma(\cdot)$, without affecting **K**?

[2] Cosme Louart, Zhenyu Liao, and Romain Couillet. "A Random Matrix Approach to Neural Networks". In: *The Annals of Applied Probability* 28.2 (2018),

Problem settings

- **Question**: what can we say on the expected kernel matrix of the two-layer NN model

$$\boxed{\mathbf{K}(\mathbf{X}) \equiv \mathbb{E}_{\mathbf{w}}[\sigma(\mathbf{X}^\mathsf{T}\mathbf{w})\sigma(\mathbf{w}^\mathsf{T}\mathbf{X})] \in \mathbb{R}^{n\times n}}$$

- and if yes, can we compress the NN by tuning **weights W** and/or **activation?** σ, without affecting **K**?

Data: K-class Gaussian mixture model (GMM)

Let $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$ be independently drawn (non-necessarily uniformly) from one of the K classes:

$$\mathcal{C}_a : \sqrt{p}\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}_a, \mathbf{C}_a), \quad a \in \{1, \ldots, K\} \tag{1}$$

Large dimensional asymptotics

As $n, p \to \infty$ with $p/n \to c \in (0, \infty)$ and some additional growth-rate assumptions on the difference $\|\boldsymbol{\mu}_a - \boldsymbol{\mu}_b\|$ and $\|\mathbf{C}_a - \mathbf{C}_b\|$, $a, b \in \{1, \ldots, K\}$.

Main result and the proof

Theorem (Asymptotic equivalent for $\mathbf{K}$, [ALC22])

For kernel matrix $\mathbf{K} = \{\mathbb{E}[\sigma(\mathbf{x}_i^\top \mathbf{w})\sigma(\mathbf{w}^\top \mathbf{x}_j)]\}_{i,j=1}^n$ *defined above, one has, as* $n, p \to \infty$ *that* $\|\mathbf{K} - \tilde{\mathbf{K}}\| \to 0$, *for some random matrix* $\tilde{\mathbf{K}}$ *dependent of data* $\mathbf{X}$, *of activation* σ *but only via the following scalars*

$$\boxed{d_0 = \mathbb{E}[\sigma^2(\sqrt{\tau}z)] - \mathbb{E}[\sigma(\sqrt{\tau}z)]^2 - \tau\mathbb{E}[\sigma'(\sqrt{\tau}z)]^2, \quad d_1 = \mathbb{E}[\sigma'(\sqrt{\tau}z)]^2, \quad d_2 = \frac{1}{4}\mathbb{E}[\sigma''(\sqrt{\tau}z)]^2}$$

and independent of the distribution of $\mathbf{W}$, *as long as of normalized to have zero mean and unit variance.*

Proof outline:

- We are interested in the kernel matrix $\mathbf{K}$, the (i,j) entry of which $\mathbf{K}_{ij} = \mathbb{E}_{\mathbf{w}}[\sigma(\mathbf{x}_i^\top \mathbf{w})\sigma(\mathbf{w}^\top \mathbf{x}_j)]$.
- Conditioned on $\mathbf{x}_i, \mathbf{x}_j$, $\mathbf{w}^\top \mathbf{x}_i \equiv \|\mathbf{x}_i\| \cdot \xi_i$ and $\mathbf{w}^\top \mathbf{x}_j$ are asymptotically Gaussian, but **correlated**!
- Gram-Schmidt to de-correlate $\mathbf{w}^\top \mathbf{x}_j = \frac{\mathbf{x}_i^\top \mathbf{x}_j}{\|\mathbf{x}_i\|}\xi_i + \sqrt{\|\mathbf{x}_j\|^2 - \frac{(\mathbf{x}_i^\top \mathbf{x}_j)^2}{\|\mathbf{x}_i\|^2}}\xi_j$, for Gaussian ξ_j now **independent** of ξ_j
- Use the fact $\mathbf{x}_i^\top \mathbf{x}_j = O(p^{-1/2})$ and $\|\mathbf{x}_i\|^2 \approx \tau/2 = O(1)$, Taylor-expand to "linearize" $\sigma(\cdot)$ to order $o(n^{-1})$
- Since $\|\mathbf{A}\|_2 \leq n\|\mathbf{A}\|_\infty$, with $\|\mathbf{A}\|_\infty = \max_{ij}|\mathbf{A}_{ij}|$, obtain **spectral** approximation $\tilde{\mathbf{K}}$.

[3] Hafiz Tiomoko Ali, Zhenyu Liao, and Romain Couillet. "Random matrices in service of ML footprint: ternary random features with no performance loss". In: *International Conference on Learning Representations*. 2022

Practical consequence of the theory

According to theorem, allowed to choose arbitrary weights $\mathbf{W}$ and activation σ, without affecting $\mathbf{K}$ asymptotically, under the following conditions:

- weights $\mathbf{W}$ have independent entries with zero mean and unit variance
- activation σ has the same few parameters as the original net

$$d_0 = \mathbb{E}[\sigma^2(\sqrt{\tau}z)] - \mathbb{E}[\sigma(\sqrt{\tau}z)]^2 - \tau\mathbb{E}[\sigma'(\sqrt{\tau}z)]^2, \quad d_1 = \mathbb{E}[\sigma'(\sqrt{\tau}z)]^2, \quad d_2 = \frac{1}{4}\mathbb{E}[\sigma''(\sqrt{\tau}z)]^2, \tag{2}$$

In particular,

- **sparse and binarized** (e.g., Bernoulli distributed) weights $\mathbf{W}$ instead of dense Gaussian weights
- **sparse quantized** (e.g., binarized) activation σ shares the same d_0, d_1, and d_2

Numerical results

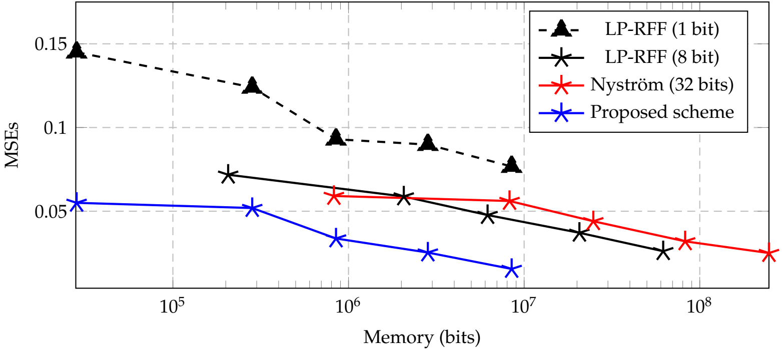


Figure: Test mean square errors of ridge regression on quantized single-hidden-layer random nets for different numbers of features $N \in \{5.10^2, 10^3, 5.10^3, 10^4, 5.10^4\}$, using LP-RFF, Nyström approximation, versus the proposed approach, on the Census dataset, with $n = 16\,000$ training samples, $n_{\text{test}} = 2\,000$ test samples, and data dimension $p = 119$.

Fully-connected deep neural networks with random weights

- everyone cares more about (i) deep neural networks and (ii) have non-random weights
- with some additional efforts, theory extends to fully-connected **deep** neural networks of depth L,

$$f(\mathbf{x}) = \frac{1}{\sqrt{d_L}} \mathbf{w}^\mathsf{T} \sigma_L \left(\frac{1}{\sqrt{d_{L-1}}} \mathbf{W}_L \sigma_{L-1} \left(\cdots \frac{1}{\sqrt{d_2}} \sigma_2 \left(\frac{1}{\sqrt{d_1}} \mathbf{W}_2 \sigma_1 (\mathbf{W}_1 \mathbf{x}) \right) \right) \right), \tag{3}$$

again for random $\mathbf{W}_1, \ldots, \mathbf{W}_L$ and activations $\sigma_1(\cdot), \ldots, \sigma_L(\cdot)$.

Theorem (Asymptotic equivalents for conjugate kernels, informal)

Under the same condition, define output features of layer $\ell \in \{1, \ldots, L\}$, as

$$\mathbf{\Sigma}_\ell = \frac{1}{\sqrt{d_\ell}} \sigma_\ell \left(\frac{1}{\sqrt{d_{\ell-1}}} \mathbf{W}_\ell \sigma_{\ell-1} \left(\cdots \frac{1}{\sqrt{d_2}} \sigma_2 \left(\frac{1}{\sqrt{d_1}} \mathbf{W}_2 \sigma_1 (\mathbf{W}_1 \mathbf{X}) \right) \right) \right). \tag{4}$$

we have for the Conjugate Kernel $\mathbf{K}_{\mathrm{CK},\ell}$ at layer ℓ defined as

$$\mathbf{K}_{\mathrm{CK},\ell} = \mathbb{E}[\mathbf{\Sigma}_\ell^\mathsf{T} \mathbf{\Sigma}_\ell] \in \mathbb{R}^{n \times n}, \tag{5}$$

that $\|\mathbf{K}_{\mathrm{CK},\ell} - \tilde{\mathbf{K}}_{\mathrm{CK},\ell}\| \to 0$, some random matrix $\tilde{\mathbf{K}}_{\mathrm{CK},\ell}$ dependent of data, of activation σ_ℓ but only via a few parameters, and independent of the distribution of $\mathbf{W}$, as long as of normalized to have zero mean and unit variance.

Theorem (Asymptotic equivalents for CK matrices, formal)

Let $\tau_0, \tau_1, \ldots, \tau_L \geq 0$ *be a sequence of non-negative numbers satisfying the following recursion:*

$$\tau_\ell = \sqrt{\mathbb{E}[\sigma_\ell^2(\tau_{\ell-1}\xi)]}, \quad \xi \sim \mathcal{N}(0,1), \quad \ell \in \{1,\ldots,L\}. \tag{6}$$

Further assume that the activation functions $\sigma_\ell(\cdot)$*s are "centered," such that* $\mathbb{E}[\sigma_\ell(\tau_{\ell-1}\xi)] = 0$*. Then, for the CK matrix* $\mathbf{K}_{\mathrm{CK},\ell}$ *of layer* $\ell \in \{1,\ldots,L\}$ *defined in* (5)*, as* $n, p \to \infty$*, one has that:*

$$\boxed{\|\mathbf{K}_{\mathrm{CK},\ell} - \tilde{\mathbf{K}}_{\mathrm{CK},\ell}\| \to 0, \quad \tilde{\mathbf{K}}_{\mathrm{CK},\ell} \equiv \alpha_{\ell,1}\mathbf{X}^\mathsf{T}\mathbf{X} + \mathbf{V}\mathbf{A}_\ell\mathbf{V}^\mathsf{T} + (\tau_\ell^2 - \tau_0^2\alpha_{\ell,1} - \tau_0^4\alpha_{\ell,3})\mathbf{I}_n,} \tag{7}$$

almost surely, with $\mathbf{V} = [\mathbf{J}/\sqrt{p},\ \boldsymbol{\psi}] \in \mathbb{R}^{n\times(K+1)}$, $\mathbf{A}_\ell = \begin{bmatrix} \alpha_{\ell,2}\mathbf{t}\mathbf{t}^\mathsf{T} + \alpha_{\ell,3}\mathbf{T} & \alpha_{\ell,2}\mathbf{t} \\ \alpha_{\ell,2}\mathbf{t}^\mathsf{T} & \alpha_{\ell,2} \end{bmatrix} \in \mathbb{R}^{(K+1)\times(K+1)}$*, for class label vectors* $\mathbf{J} = [\mathbf{j}_1, \ldots, \mathbf{j}_K] \in \mathbb{R}^{n\times K}$*, "second-order" data fluctuation vector* $\boldsymbol{\psi} \in \mathbb{R}^n$*, second-order data statistics* $\mathbf{t} = \{\operatorname{tr} \mathbf{C}_a^\circ/\sqrt{p}\}_{a=1}^K \in \mathbb{R}^K$ *and* $\mathbf{T} = \{\operatorname{tr} \mathbf{C}_a\mathbf{C}_b/p\}_{a,b=1}^K \in \mathbb{R}^{K\times K}$*, as well as non-negative* $\alpha_{\ell,1}, \alpha_{\ell,2}, \alpha_{\ell,3}$ *satisfying*

$$\alpha_{\ell,1} = \mathbb{E}[\sigma_\ell'(\tau_{\ell-1}\xi)]^2\alpha_{\ell-1,1}, \quad \alpha_{\ell,2} = \mathbb{E}[\sigma_\ell'(\tau_{\ell-1}\xi)]^2\alpha_{\ell-1,2} + \frac{1}{4}\mathbb{E}[\sigma_\ell''(\tau_{\ell-1}\xi)]^2\alpha_{\ell-1,4}^2, \tag{8}$$

$$\alpha_{\ell,3} = \mathbb{E}[\sigma_\ell'(\tau_{\ell-1}\xi)]^2\alpha_{\ell-1,3} + \frac{1}{2}\mathbb{E}[\sigma_\ell''(\tau_{\ell-1}\xi)]^2\alpha_{\ell-1,1}^2. \tag{9}$$

with $\alpha_{\ell,4} = \mathbb{E}\left[(\sigma_\ell'(\tau_{\ell-1}\xi))^2 + \sigma_\ell(\tau_{\ell-1}\xi)\sigma_\ell''(\tau_{\ell-1}\xi)\right]\alpha_{\ell-1,4}$ *for* $\xi \sim \mathcal{N}(0,1)$*.*

Deep compression of fully-connected deep nets via NTK

- used for compression of fully-connected deep nets, but with **random** weights only, who cares?
- **Our approach**: from random to trained nets via Neural Tangent Kernel (NTK) theory [JGH18]:
- for **(i)** sufficiently wide nets **(ii)** trained with gradient descent of sufficiently small step size, NTK is **determined** at random initialization and remains unchanged during training

Proof outline of NTK

– conditioned on $(\mathbf{x}_i, y_i)_{i=1}^n$, train NN by minimizing $\ell(\theta) = \frac{1}{2}\sum_{i=1}^n (f(\theta, \mathbf{x}_i) - y_i)^2$, $\theta \equiv \{\mathbf{w}, \mathbf{W}_L, \cdots, \mathbf{W}_1\}$;

– gradient descent with sufficiently small step size leads to gradient flow dynamics: $\frac{d\theta(t)}{dt} = -\nabla \ell(\theta(t))$;

– the dynamics of the output vector $\mathbf{u}(t) \in \mathbb{R}^n$ with $u_i = \frac{df(\theta(t), \mathbf{x}_i)}{dt}$ given by

$$\frac{d\mathbf{u}(t)}{dt} = -\hat{\mathbf{K}}_{\rm NTK}(t)\,(\mathbf{u}(t) - \mathbf{y})\,, \quad \mathbf{y} = [y_1, \ldots, y_n]^{\sf T}, \quad [\hat{\mathbf{K}}_{\rm NTK}(t)]_{i,j} = \left\langle \frac{\partial f(\theta, \mathbf{x}_i)}{\partial \theta}, \frac{\partial f(\theta, \mathbf{x}_j)}{\partial \theta} \right\rangle \tag{10}$$

– then, **step (1)**: convergence of the random NTK to its expectation $\hat{\mathbf{K}}_{\rm NTK}(t=0) \to \mathbf{K}_{\rm NTK} \equiv \mathbb{E}[\hat{\mathbf{K}}_{\rm NTK}(t=0)]$, and **step (2)**: stability of the NTK during training $\hat{\mathbf{K}}_{\rm NTK}(t) \simeq \hat{\mathbf{K}}_{\rm NTK}(t=0) \simeq \mathbf{K}_{\rm NTK}$ for $t > 0$.

- with some additional efforts, **understand** the behavior of NTK matrices $\mathbf{K}_{\rm NTK}$
- use the theory for DNN compression!

[4] Arthur Jacot, Franck Gabriel, and Clément Hongler. "Neural tangent kernel: Convergence and generalization in neural networks". In: *Advances in neural information processing systems*. 2018, pp. 8571–8580

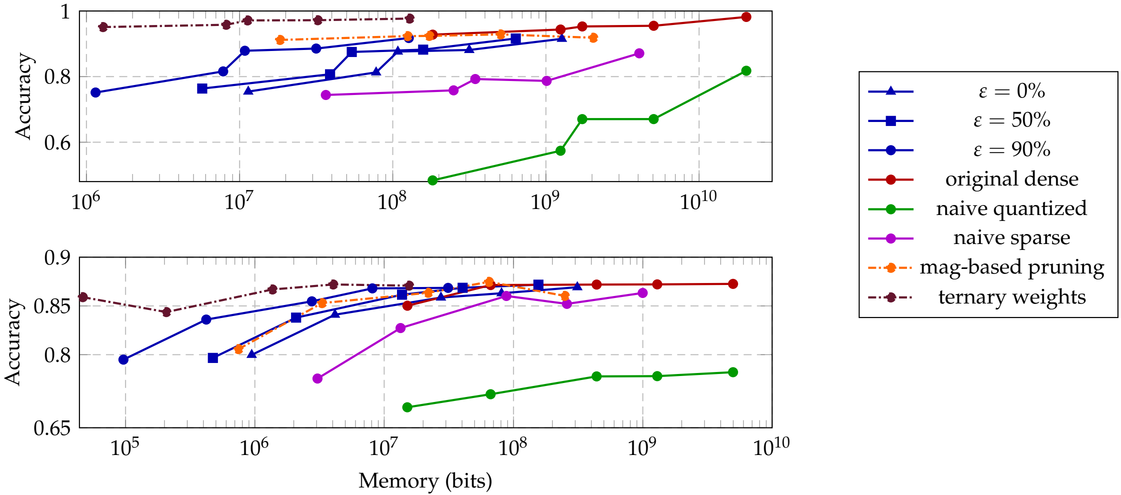


Figure: Test accuracies of compressed nets on MNIST (**top**) and CIFAR10 (**bottom**) datasets. **Blue** represent the proposed approach with different sparsity levels, **purple** represent the heuristic sparsification approach by uniformly zeroing out 80% of the weights, **green** represent the heuristic quantization approach using the binary activation $\sigma(t) = 1_{t<-1} + 1_{t>1}$, **red** represent the original network, **brown** represent the proposed compression approach *without* activation quantization, and **orange** represent magnitude-based pruning. Memory varies due to the **change of layer width** of the network.

Conclusion and take-away message

Take-away message:

- theoretical analysis of single-hidden-layer NN with random weights
- extension to fully-connected **deep** nets and to NTK
- to propose DNN compression approach with **theoretical guarantee**!

Future work and open problems:

- deep learning theory beyond the NTK regime? more challenging due to optimization and complicated dependent structure therein;
- RMT for more **structured data**, e.g., structured random graph (dense and sparse), with application in computer science
- **RMT+OPT**: RMT and high-dimensional statistics for optimization beyond worst-case scenario

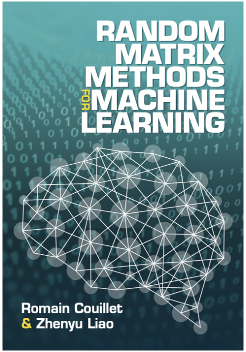
References:

- Hafiz Tiomoko Ali, Zhenyu Liao, and Romain Couillet. "Random matrices in service of ML footprint: ternary random features with no performance loss". In: *International Conference on Learning Representations*. 2022
- Lingyu Gu et al. ""Lossless" Compression of Deep Neural Networks: A High-dimensional Neural Tangent Kernel Approach". In: *Advances in Neural Information Processing Systems*. 2022

RMT for machine learning: from theory to practice

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!



- *Random Matrix Methods for Machine Learning*, Cambridge University Press, 2022
- by Romain Couillet and **Zhenyu Liao**
- a pre-production version of the book and exercise solutions at `https://zhenyu-liao.github.io/book/`
- MATLAB and Python codes to reproduce all figures at `https://github.com/Zhenyu-LIAO/RMT4ML`

Thank you! Q & A?