A Random Matrix Approach to Explicit and Implicit Deep Neural Networks

@ Institut de Mathématiques de Toulouse, France

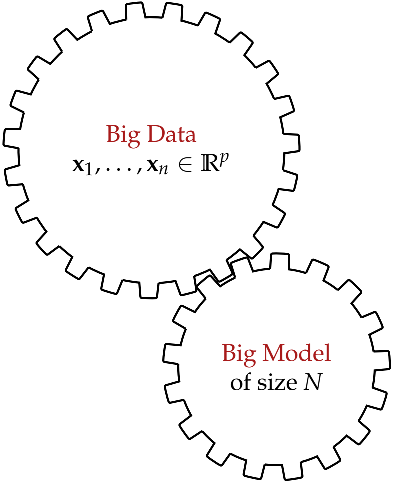
Zhenyu Liao

School of Electronic Information and Communications
Huazhong University of Science and Technology

July 2nd, 2024

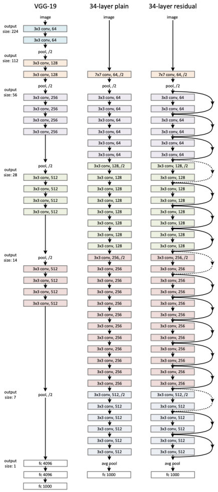
Outline

Motivation: understanding large-dimensional machine learning



- **Big Data era**: exploit large n, p, N
- counterintuitive phenomena different from classical asymptotics statistics
- complete change of understanding of many methods in statistics and machine learning
- **Random Matrix Theory (RMT)** provides the tools!
- In this talk, a RMT approach to explicit and implicit **deep neural networks (DNNs)**, with applications to DNN model compression.

Question: what are deep neural networks?



Deep Learning (DL) ≈ **multilayered neural network** (NN) is becoming the most popular machine learning (ML) model, but

- what is machine learning?
- what is a deep neural network (DNN)?
- how is such as network trained?
- is there any theory for DL, and if yes, how far is the theory from practice?

Credit: most materials in this part are borrowed from [HH19].

[1] Catherine F. Higham and Desmond J. Higham. "Deep Learning: An Introduction for Applied Mathematicians". In: *SIAM Review* 61.4 (Jan. 2019), pp. 860–891

Example: binary classification of points in $\mathbb{R}^2$

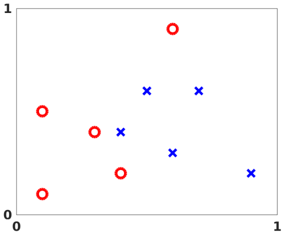


Figure: Labeled data points $\mathbf{x} \in \mathbb{R}^2$. Circles denote points in class $\mathcal{C}_1$. Crosses denote points in class $\mathcal{C}_2$.

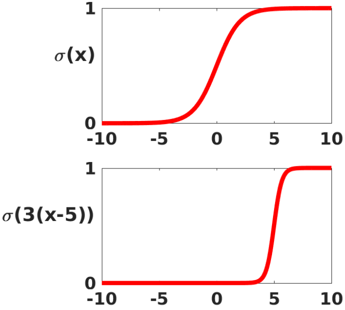


Figure: Sigmoid function.

- build a model/**function** f (from above historical data) that takes any points $\mathbf{x} \in \mathbb{R}^2$ and returns $\mathcal{C}_1$ or $\mathcal{C}_2$
- **logistic regression**: $\boxed{f(\mathbf{x}) = \sigma(\mathbf{w}^\mathsf{T}\mathbf{x} + b)}$ for $\mathbf{w} \in \mathbb{R}^2$ and $b \in \mathbb{R}$ to be determined, and sigmoid function $\sigma(t) = \frac{1}{1+e^{-t}}$
- "learn" or estimate parameters $\mathbf{w}, b$ from data/samples, by minimizing some cost function (e.g., negative likelihood, MSE)
- predict $\mathbf{x} \in \mathcal{C}_1$ if $f(\mathbf{x}) < 1/2$ and $\mathbf{x} \in \mathcal{C}_2$ otherwise.

Neural networks are nothing but "cascaded" logistic regressors

- logistic regression $f(\mathbf{x}) = \sigma(\mathbf{w}^\mathsf{T}\mathbf{x} + b) \in \mathbb{R}$ for $\mathbf{w} \in \mathbb{R}^2$, $b \in \mathbb{R}$ extends to

$$\boxed{f(\mathbf{x}) = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b}) \in \mathbb{R}^N} \quad \mathbf{W} \in \mathbb{R}^{N\times 2}, \mathbf{b} \in \mathbb{R}^N \tag{1}$$

  and $\sigma(\cdot)$ applied entry-wise: this is **one layer** of a DNN
- repeat this to make the network deep, with possibly different width in each layer

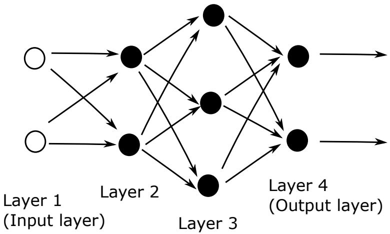


Figure: A network with four layers.

- $\sigma(\mathbf{W}_2 x + \mathbf{b}_2) \in \mathbb{R}^2$, $\sigma\left(\mathbf{W}_3\sigma(\mathbf{W}_2\mathbf{x} + \mathbf{b}_2) + \mathbf{b}_3\right) \in \mathbb{R}^3$
- $f_{4L-NN}(\mathbf{x}) = \sigma\left(\mathbf{W}_4\sigma\left(\mathbf{W}_3\sigma(\mathbf{W}_2\mathbf{x} + \mathbf{b}_2) + \mathbf{b}_3\right) + \mathbf{b}_4\right) \in \mathbb{R}^2$

Define the label/target output as

$$\mathbf{y}(\mathbf{x}_i) = \begin{cases} \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \mathbf{x}_i \in \mathcal{C}_1, \\ \begin{bmatrix} 0 \\ 1 \end{bmatrix} & \mathbf{x}_i \in \mathcal{C}_2. \end{cases} \tag{2}$$

the MSE cost function writes $\text{Cost}\left(\mathbf{W}_2, \mathbf{W}_3, \mathbf{W}_4, \mathbf{b}_2, \mathbf{b}_3, \mathbf{b}_4\right) = \frac{1}{10}\sum_{i=1}^{10} \|\mathbf{y}(\mathbf{x}_i) - f_{4L-NN}(\mathbf{x}_i)\|^2$

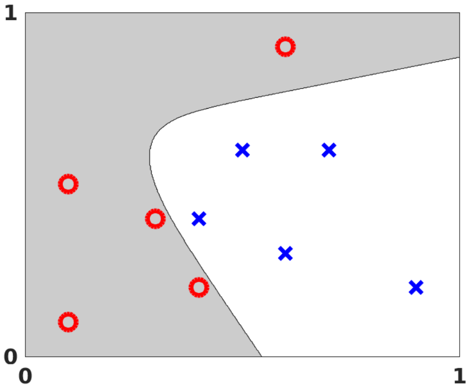


Figure: Visualization of output from a multilayered neural network applied to the data.

- from training to test!

General formulation and gradient decent training of DNN

We can define the network in a **layer-by-layer** fashion:

$$\mathbf{a}_0 = \mathbf{x} \in \mathbb{R}^{N_0}, \quad \boxed{\mathbf{a}_\ell = \sigma\left(\mathbf{W}_\ell \mathbf{a}_{\ell-1} + \mathbf{b}_\ell\right)} \in \mathbb{R}^{N_\ell}, \quad \ell = 1, \ldots, L,$$

with weights $\mathbf{W}_\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ and bias $\mathbf{b} \in \mathbb{R}^{N_\ell}$ at layer ℓ.

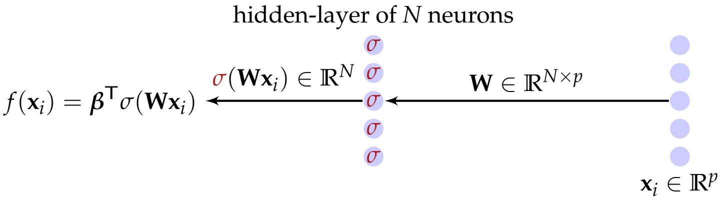
- $\mathbf{W}_\ell$s and $\mathbf{b}_\ell$s obtained by minimizing cost function on a given training set $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ of size n:

$$\text{Cost} = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2} \|\mathbf{y}_i - \mathbf{a}_L(\mathbf{x}_i)\|^2. \tag{3}$$

- update using (stochastic) gradient descent, for parameter P,

$$P(t+1) = P(t) - \eta \nabla_P \text{Cost}(P(t)). \tag{4}$$

Two-layer network with random first layer



- for random (first-layer) weights $\mathbf{W} \in \mathbb{R}^{N\times p}$ having say i.i.d. standard Gaussian entries
- get second-layer $\boldsymbol{\beta}$ by minimizing Cost $= \frac{1}{n}\sum_{i=1}^n (y_i - \boldsymbol{\beta}^\mathsf{T}\sigma(\mathbf{W}\mathbf{x}_i))^2 + \gamma\|\boldsymbol{\beta}\|^2$ for some regularization parameter $\gamma > 0$, then

$$\boldsymbol{\beta} \equiv \frac{1}{n}\boldsymbol{\Sigma}\left(\frac{1}{n}\boldsymbol{\Sigma}^\mathsf{T}\boldsymbol{\Sigma} + \gamma\mathbf{I}_n\right)^{-1}\mathbf{y}, \tag{5}$$

- training MSE (on the given training set $(\mathbf{X},\mathbf{y})$) reads

$$E_{\text{train}} = \frac{1}{n}\|\mathbf{y} - \boldsymbol{\Sigma}^\mathsf{T}\boldsymbol{\beta}\|_F^2 = \frac{\gamma^2}{n}\mathbf{y}\mathbf{Q}^2(\gamma)\mathbf{y}, \quad \boxed{\mathbf{Q}(\gamma) \equiv \left(\frac{1}{n}\boldsymbol{\Sigma}^\mathsf{T}\boldsymbol{\Sigma} + \gamma\mathbf{I}_n\right)^{-1}} \tag{6}$$

- Similarly, the test MSE on a test set $(\hat{\mathbf{X}},\hat{\mathbf{y}}) \in \mathbb{R}^{p\times\hat{n}} \times \mathbb{R}^{d\times\hat{n}}$ of size $\hat{n}$: $E_{\text{test}} = \frac{1}{\hat{n}}\|\hat{\mathbf{y}} - \hat{\boldsymbol{\Sigma}}^\mathsf{T}\boldsymbol{\beta}\|_F^2$, $\quad \hat{\boldsymbol{\Sigma}} = \sigma(\mathbf{W}\hat{\mathbf{X}})$.

Study of CK in the infinite-neuron regime

- **Key object**: empirical CK $\frac{1}{N}\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma}$, correlation in the feature space, for random initialization: $\mathbf{W}_{ij} \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$, relates to linearized model $f_{\rm lin}$
- $\frac{1}{N}\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma} = \frac{1}{N}\sum_{i=1}^N \sigma(\mathbf{X}^\mathsf{T}\mathbf{w}_i)\sigma(\mathbf{w}_i^\mathsf{T}\mathbf{X})$ for independent $\mathbf{w}_i \sim \mathcal{N}(\mathbf{0},\mathbf{I}_p)$.
- In the **infinite-neuron limit** ($N \to \infty$), convergence to the limiting CK matrix

$$\boxed{\frac{1}{N}\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma} \to \mathbf{K}_{\rm CK}(\mathbf{X}) \equiv \mathbb{E}_{\mathbf{w}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_p)}[\sigma(\mathbf{X}^\mathsf{T}\mathbf{w})\sigma(\mathbf{w}^\mathsf{T}\mathbf{X})] \in \mathbb{R}^{n\times n}}$$

- theoretical **understanding** of NN model: generalization? optimization?
- **Application**: compress NN by carefully choosing **weights W** and/or **activation?** σ, e.g., without changing $\mathbf{K}_{\rm CK}$?

Problem settings

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:

$$\mathcal{C}_a : \sqrt{p}\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}_a, \mathbf{C}_a), \quad a \in \{1, \ldots, K\} \tag{7}$$

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 $\|\mathbf{C}_a - \mathbf{C}_b\|$, $a, b \in \{1, \ldots, K\}$, as $n, p \to \infty$.

Theorem (Asymptotic approximation for conjugate kernels, [AZC22])

For CK matrix $\mathbf{K}_{\rm CK} = \{\mathbb{E}[\sigma(\mathbf{x}_i^\mathsf{T}\mathbf{w})\sigma(\mathbf{w}^\mathsf{T}\mathbf{x}_j)]\}_{i,j=1}^n$ *defined above, one has, as* $n, p \to \infty$ *that* $\|\mathbf{K}_{\rm CK} - \tilde{\mathbf{K}}_{\rm CK}\| \to 0$, *for some random matrix* $\tilde{\mathbf{K}}_{\rm CK}$ *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.*

Main result and the proof

Theorem (Asymptotic approximation for conjugate kernels, [AZC22])

For CK matrix $\mathbf{K}_{\rm CK} = \{\mathbb{E}[\sigma(\mathbf{x}_i^\mathsf{T}\mathbf{w})\sigma(\mathbf{w}^\mathsf{T}\mathbf{x}_j)]\}_{i,j=1}^n$ defined above, one has, as $n,p \to \infty$ that $\|\mathbf{K}_{\rm CK} - \tilde{\mathbf{K}}_{\rm CK}\| \to 0$, for some random matrix $\tilde{\mathbf{K}}_{\rm CK}$ 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 sketch:

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

[2] 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 (ICLR 2022)*. 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

$$d_0 = \mathbb{E}[\sigma^2(\sqrt{\tau} z)] - \mathbb{E}[\sigma(\sqrt{\tau} z)]^2 - \tau \mathbb{E}[\sigma'(\sqrt{\tau} z)]^2, \quad d_1 = \mathbb{E}[\sigma'(\sqrt{\tau} z)]^2, \quad d_2 = \frac{1}{4}\mathbb{E}[\sigma''(\sqrt{\tau} z)]^2, \tag{8}$$

In particular,

- **sparse and binarized** (e.g., Bernoulli distributed) weights $\mathbf{W}$ instead of dense Gaussian weights

$$[\mathbf{W}]_{ij} = 0 \text{ with proba } \varepsilon \in [0,1), \quad [\mathbf{W}]_{ij} = \pm(1-\varepsilon)^{-1/2} \text{ each with proba } 1/2 - \varepsilon/2, \tag{9}$$

- **sparse quantized** (e.g., binarized) activation σ shares the same d_0, d_1, and d_2

Numerical results

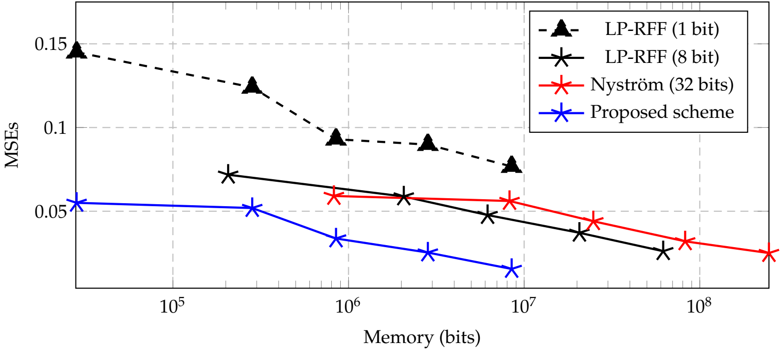


Figure: Test mean square errors of ridge regression on quantized single-hidden-layer random nets for different numbers of features $N \in \{5.10^2, 10^3, 5.10^3, 10^4, 5.10^4\}$, using LP-RFF, Nyström approximation, versus the proposed approach, on the Census dataset, with $n = 16\,000$ training samples,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

CK of fully-connected deep neural networks

- everyone cares more about deep neural networks
- with some additional efforts, theory extends to fully-connected **deep** neural networks of depth L,

$$f(\mathbf{x}) = \frac{1}{\sqrt{d_L}} \mathbf{w}^\mathsf{T} \sigma_L \left(\frac{1}{\sqrt{d_{L-1}}} \mathbf{W}_L \sigma_{L-1} \left(\cdots \frac{1}{\sqrt{d_2}} \sigma_2 \left(\frac{1}{\sqrt{d_1}} \mathbf{W}_2 \sigma_1 (\mathbf{W}_1 \mathbf{x}) \right) \right) \right), \tag{10}$$

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

Theorem (Asymptotic approximation for conjugate kernels, informal)

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

$$\mathbf{\Sigma}_\ell = \frac{1}{\sqrt{d_\ell}} \sigma_\ell \left(\frac{1}{\sqrt{d_{\ell-1}}} \mathbf{W}_\ell \sigma_{\ell-1} \left(\cdots \frac{1}{\sqrt{d_2}} \sigma_2 \left(\frac{1}{\sqrt{d_1}} \mathbf{W}_2 \sigma_1 (\mathbf{W}_1 \mathbf{X}) \right) \right) \right). \tag{11}$$

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

$$\mathbf{K}_{\mathrm{CK},\ell} = \mathbb{E}[\mathbf{\Sigma}_\ell^\mathsf{T} \mathbf{\Sigma}_\ell] \in \mathbb{R}^{n \times n}, \tag{12}$$

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 approximation for CK matrices, formal, [Gu+22])

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

$$\tau_\ell = \sqrt{\mathbb{E}[\sigma_\ell^2(\tau_{\ell-1}\xi)]}, \quad \xi \sim \mathcal{N}(0,1), \quad \ell \in \{1, \ldots, L\}. \tag{13}$$

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* (12), *as* $n, p \to \infty$, *one has that:*

$$\boxed{\|\mathbf{K}_{\mathrm{CK},\ell} - \tilde{\mathbf{K}}_{\mathrm{CK},\ell}\| \to 0, \quad \tilde{\mathbf{K}}_{\mathrm{CK},\ell} \equiv \alpha_{\ell,1}\mathbf{X}^\mathsf{T}\mathbf{X} + \mathbf{V}\mathbf{A}_\ell\mathbf{V}^\mathsf{T} + (\tau_\ell^2 - \tau_0^2\alpha_{\ell,1})\mathbf{I}_n,} \tag{14}$$

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*

$$\alpha_{\ell,1} = \mathbb{E}[\sigma_\ell'(\tau_{\ell-1}\xi)]^2\alpha_{\ell-1,1}, \quad \alpha_{\ell,2} = \mathbb{E}[\sigma_\ell'(\tau_{\ell-1}\xi)]^2\alpha_{\ell-1,2} + \frac{1}{4}\mathbb{E}[\sigma_\ell''(\tau_{\ell-1}\xi)]^2\alpha_{\ell-1,4}^2, \tag{15}$$

$$\alpha_{\ell,3} = \mathbb{E}[\sigma_\ell'(\tau_{\ell-1}\xi)]^2\alpha_{\ell-1,3} + \frac{1}{2}\mathbb{E}[\sigma_\ell''(\tau_{\ell-1}\xi)]^2\alpha_{\ell-1,1}^2. \tag{16}$$

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* $\xi \sim \mathcal{N}(0,1)$.

Fully-connected deep nets: CK, NTK, and beyond

- happy with the study of (limiting) CK for DNN models
- extension to NTK via intrinsic **connection** between CK and NTK [JGH18]

$$\mathbf{K}_{\mathrm{NTK},\ell}(\mathbf{X}) = \mathbf{K}_{\mathrm{CK},\ell}(\mathbf{X}) + \mathbf{K}_{\mathrm{NTK},\ell-1}(\mathbf{X}) \circ \mathbf{K}'_{\mathrm{CK},\ell}(\mathbf{X}), \quad \mathbf{K}_{\mathrm{NTK},0}(\mathbf{X}) = \mathbf{K}_{\mathrm{CK},0}(\mathbf{X}) = \mathbf{X}^\mathsf{T}\mathbf{X}, \tag{17}$$

  and some additional efforts
- **convergence** and **generalization** theory via NTK [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, and applies to **explicitly** characterize DNN convergence and generalization properties
- we can use the theory for DNN compression!

[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

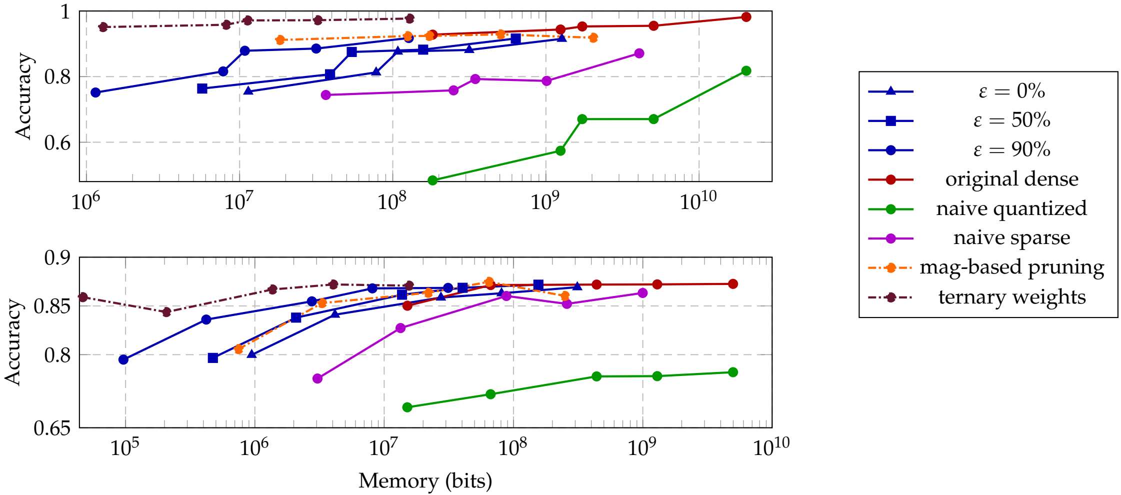


Figure: Test accuracy of classification on MNIST (**top**) and CIFAR10 (**bottom**) datasets. **Blue**: proposed NTK-LC approach with different levels of sparsity $\varepsilon \in \{0\%, 50\%, 90\%\}$, **purple**: heuristic sparsification approach by uniformly zeroing out 80% of the weights, **green**: heuristic quantization approach with binary activation $\sigma(t) = 1_{t<-1} + 1_{t>1}$, **red**: original network, **orange**: NTK-LC *without* activation quantization, and **brown**: magnitude-based pruning with same sparsity level as **orange**. Memory varies due to the **change of layer width** of the network.

Deep equilibrium model (DEQ), [BKK19]

Let $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in \mathbb{R}^{p \times n}$ denote the input data, consider a vanilla DEQ with output $f(\mathbf{x}_i)$ given by

$$f(\mathbf{x}_i) = \boldsymbol{\beta}^\mathsf{T} \mathbf{z}_i^*, \tag{18}$$

where $\boldsymbol{\beta} \in \mathbb{R}^m$ and $\mathbf{z}_i^{(*)} \equiv \lim_{l \to \infty} \mathbf{z}_i^{(l)} \in \mathbb{R}^m$ with

$$\mathbf{z}_i^{(l)} = \frac{1}{\sqrt{m}} \phi \left(\sigma_a \mathbf{A} \mathbf{z}_i^{(l-1)} + \sigma_b \mathbf{B} \mathbf{x}_i \right) \in \mathbb{R}^m, \text{ for } l \geq 1, \tag{19}$$

for some appropriate initialization $\mathbf{z}_i^{(0)}$, $\mathbf{A} \in \mathbb{R}^{m \times m}$ and $\mathbf{B} \in \mathbb{R}^{m \times p}$ are DEQ weights, $\sigma_a, \sigma_b \in \mathbb{R}$ are constants, and ϕ is an element-wise activation. Note $\mathbf{z}_i^*$ can also be determined as the equilibrium point of

$$\mathbf{z}_i^* = \frac{1}{\sqrt{m}} \phi \left(\sigma_a \mathbf{A} \mathbf{z}_i^* + \sigma_b \mathbf{B} \mathbf{x}_i \right). \tag{20}$$

[4] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. "Deep Equilibrium Models". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019

Connection between Implicit and Explicit NNs

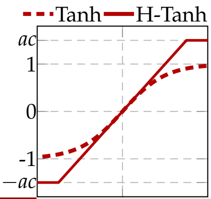
- similar analysis can be performed for such Implicit-NN models as well
- leads to high-dimensional "**equivalence**" (in the sense of CK or NTK) between **Implicit** and **Explicit** NNs

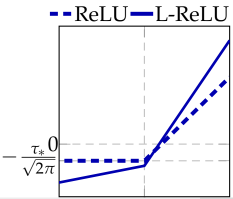
Theorem (Asymptotic approximation for Implicit-CK matrices)

For the DEQ model under study, under some mild technical assumptions, and let the activation ϕ *be centered such that* $\mathbb{E}[\phi(\tau_* \xi)] = 0$ *for* $\xi \sim \mathcal{N}(0,1)$ *and* τ_* *be such that* $\tau_* = \sqrt{\sigma_a^2 \mathbb{E}\left[\phi^2(\tau_* \xi)\right] + \sigma_b^2 \tau_0^2}$. *Then, the Implicit-CK matrix* $\mathbf{G}^*$ *satisfies* $\|\mathbf{G}^* - \overline{\mathbf{G}}\| \to 0$ *almost surely as* $n, p \to \infty$, *for a random matrix* $\overline{\mathbf{G}}$ *explicitly given by*

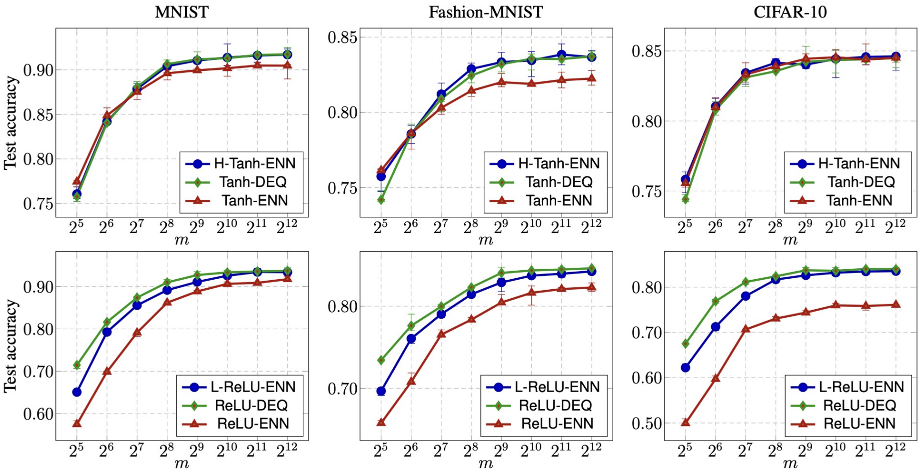
$$\overline{\mathbf{G}} \equiv \alpha_{*,1} \mathbf{X}^\mathsf{T} \mathbf{X} + \mathbf{V} \mathbf{C}_* \mathbf{V}^\mathsf{T} + ({\gamma_*}^2 - \tau_0^2 \alpha_{*,1}) \mathbf{I}_n, \quad \mathbf{C}_* = \begin{bmatrix} \alpha_{*,2} \mathbf{t}\mathbf{t}^\mathsf{T} + \alpha_{*,3} \mathbf{T} & \alpha_{*,2} \mathbf{t} \\ \alpha_{*,2} \mathbf{t}^\mathsf{T} & \alpha_{*,2} \end{bmatrix} \in \mathbb{R}^{(K+1) \times (K+1)} \tag{21}$$

for explicit parameters $\gamma_*, \alpha_{*,1}, \alpha_{*,2}, \alpha_{*,3} \geq 0$.





Numerical results



Take-away messages:

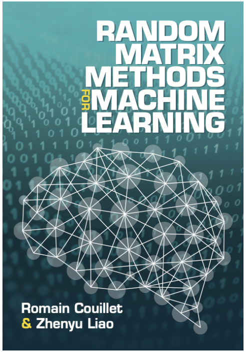
- for GMM input data, RMT allows for precise characterization of (the CKs of) **random** shallow and deep neural networks
- extends to **NTKs**, providing access to trained DNNs, but **only** in the "lazy" NTK regime
- makes explicit connections between **Implicit** and **Explicit** NNs

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 (ICLR 2022)*. 2022
- Lingyu Gu, Yongqi Du, Yuan Zhang, Di Xie, Shiliang Pu, Robert Qiu, and **Zhenyu Liao**. ""Lossless" Compression of Deep Neural Networks: A High-dimensional Neural Tangent Kernel Approach". In: *Advances in Neural Information Processing Systems*. Vol. 35. Curran Associates, Inc., 2022, pp. 3774–3787 (**Please** refer to the ArXiv version on `https://arxiv.org/abs/2403.00258` that fixed typos in Theorems 1 and 2 from the NeurIPS 2022 proceeding version.)
- Z. Ling, L. Li, Z. Feng, Y. Zhang, F. Zhou, R. C. Qiu, **Z. Liao** "Deep Equilibrium Models are Almost Equivalent to Not-so-deep Explicit Models for High-dimensional Gaussian Mixtures", The Forty-first International Conference on Machine Learning (**ICML 2024**), 2024

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!



- book "*Random Matrix Methods for Machine Learning*"
- by Romain Couillet and **Zhenyu Liao**
- Cambridge University Press, 2022
- 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?