Gaussian Equivalence for Nonlinear Random Matrices: Why it Works—and Where it Fails

Yue M. Lu

Harvard University

August 9, 2025

Classical random matrix models

Data matrix:

$$X = \underbrace{\begin{bmatrix} | & | & & | \\ x_1 & x_2 & \dots & x_n \\ | & | & & | \end{bmatrix}}_{n} \left.\vphantom{\begin{bmatrix} | \\ x \\ | \end{bmatrix}}\right\} d$$

Centered random vectors $x_1, \dots, x_n \overset{\text{iid}}{\sim} p(x)$

Classical random matrix models

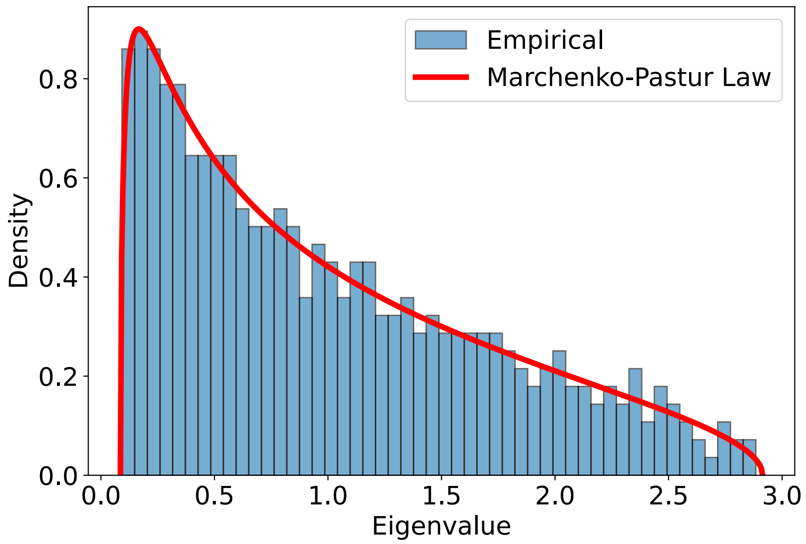
Data matrix:

$$X = \underbrace{\begin{bmatrix} | & | & & | \\ x_1 & x_2 & \dots & x_n \\ | & | & & | \end{bmatrix}}_{n} \left.\vphantom{\begin{bmatrix} | \\ x \\ | \end{bmatrix}}\right\} d$$

Centered random vectors $x_1, \dots, x_n \overset{\text{iid}}{\sim} p(x)$

Sample covariance: $H = XX^\top$ (or the ***Gram matrix*** $E = X^\top X$)

Spectrum of H $\xrightarrow[n/d \to \alpha]{d, n \to \infty}$ ***Marchenko-Pastur law***



[Marchenko & Pastur ’67], [Silverstein & Bai, ’95], [Erdos et al., ’12]

This lecture: Nonlinear random matrix ensembles

Kernel method:

$$K_{ij} = \sigma(\|x_i - x_j\|^2)$$

$\sigma(\cdot)$: elementwise nonlinear function

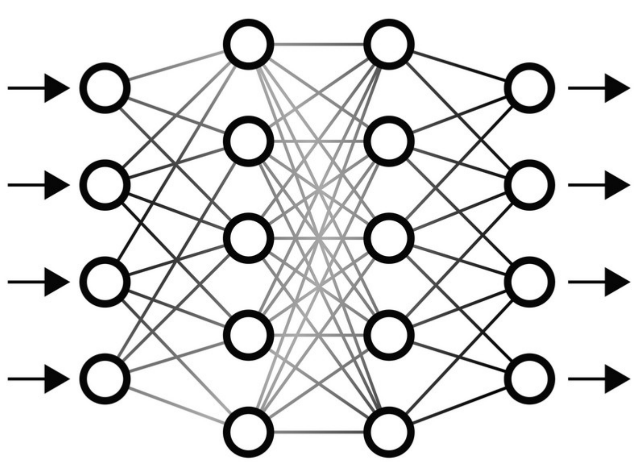
This lecture: Nonlinear random matrix ensembles

Kernel method:

$$K_{ij} = \sigma(\|x_i - x_j\|^2)$$

$\sigma(\cdot)$: elementwise nonlinear function

Multilayer neural networks:



$$\sigma(W_3\sigma(W_2\sigma(W_1X)))$$

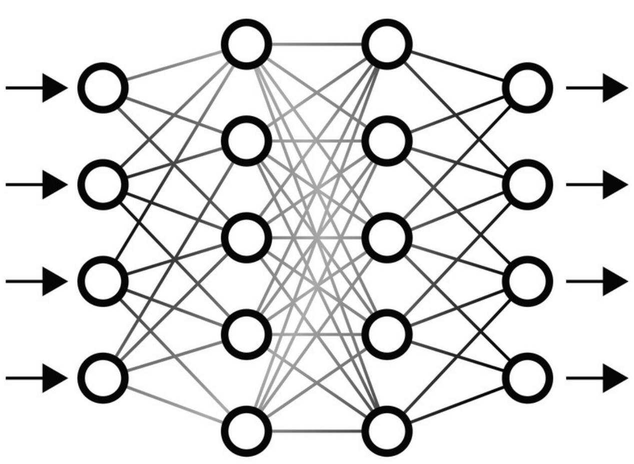
This lecture: Nonlinear random matrix ensembles

Kernel method:

$$K_{ij} = \sigma(\|x_i - x_j\|^2)$$

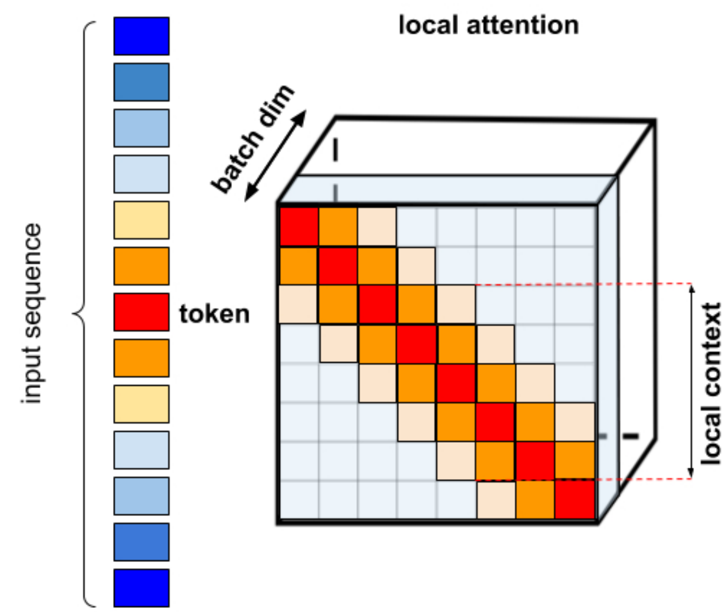
$\sigma(\cdot)$: elementwise nonlinear function

Multilayer neural networks:



$$\sigma(W_3\sigma(W_2\sigma(W_1X)))$$

Attention and Transformers:



$$\text{Softmax}(X^\top W X)$$

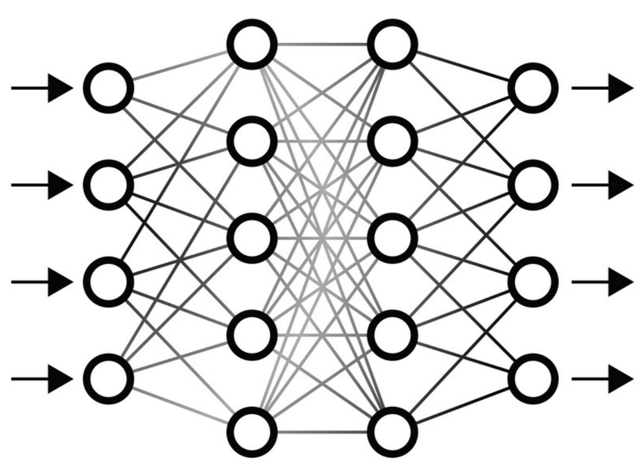
This lecture: Nonlinear random matrix ensembles

Kernel method:

$$K_{ij} = \sigma(\|x_i - x_j\|^2)$$

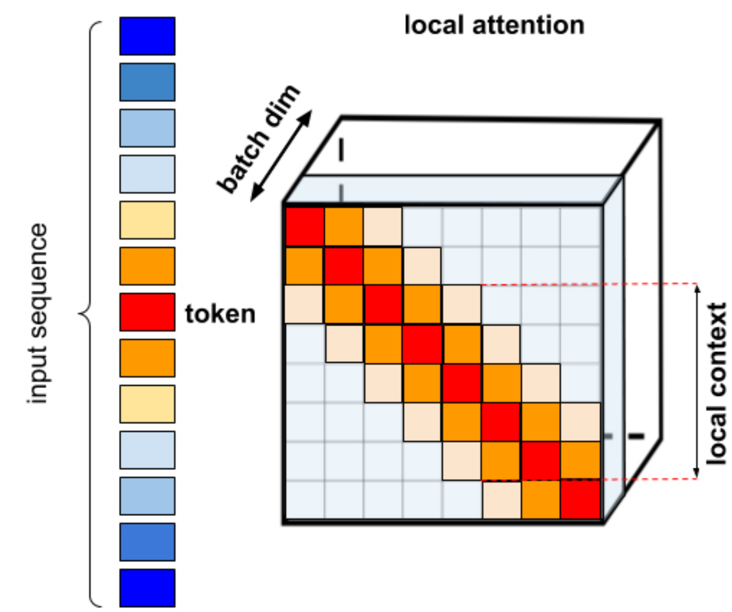
$\sigma(\cdot)$: elementwise nonlinear function

Multilayer neural networks:



$$\sigma(W_3\sigma(W_2\sigma(W_1X)))$$

Attention and Transformers:



$$\text{Softmax}(X^\top W X)$$

Other applications:

- Shrinkage of covariance matrices
- Nonlinear matrix factorization
- ...

The simpler regime

Example: Inner product kernel matrix [El Karoui '10]

$$H = (h_{ij})_{i,j \le n} = X^\top X \qquad\qquad A = \sigma(H)$$

Standard random matrix scaling: $h_{ij} = \mathcal{O}_p(1/\sqrt{n})$ for $i \neq j$

The simpler regime

Example: Inner product kernel matrix [El Karoui '10]

$$H = (h_{ij})_{i,j \leq n} = X^\top X \qquad\qquad A = \sigma(H)$$

Standard random matrix scaling: $h_{ij} = \mathcal{O}_p(1/\sqrt{n})$ for $i \neq j$

Taylor series expansion:

$$A_{ij} = \sigma(h_{ij}) = \sigma(0) + \sigma'(0)h_{ij} + \frac{\sigma''(0)}{2}h_{ij}^2 + \mathrm{H.O.T.}$$

The simpler regime

Example: Inner product kernel matrix [El Karoui '10]

$$H = (h_{ij})_{i,j \le n} = X^\top X \qquad\qquad A = \sigma(H)$$

Standard random matrix scaling: $h_{ij} = \mathcal{O}_p(1/\sqrt{n})$ for $i \neq j$

Taylor series expansion:

negligible

$$A_{ij} = \sigma(h_{ij}) = \sigma(0) + \sigma'(0) h_{ij} + \frac{\sigma''(0)}{2} h_{ij}^2 + \text{H.O.T.}$$

The simpler regime

Example: Inner product kernel matrix [El Karoui '10]

$$H = (h_{ij})_{i,j \le n} = X^\top X \qquad\qquad A = \sigma(H)$$

Standard random matrix scaling: $h_{ij} = \mathcal{O}_p(1/\sqrt{n})$ for $i \neq j$

Taylor series expansion:

negligible

$$A_{ij} = \sigma(h_{ij}) = \sigma(0) + \sigma'(0) h_{ij} + \frac{\sigma''(0)}{2} h_{ij}^2 + \mathrm{H.O.T.}$$

This part (after centering) is also negligible.
operator norm $= \mathcal{O}_p(d^{-1/2})$

The simpler regime

Example: Inner product kernel matrix [El Karoui '10]

$$H = (h_{ij})_{i,j \le n} = X^\top X \qquad\qquad A = \sigma(H)$$

Standard random matrix scaling: $h_{ij} = \mathcal{O}_p(1/\sqrt{n})$ for $i \neq j$

Taylor series expansion:

negligible

$$A_{ij} = \sigma(h_{ij}) = \sigma(0) + \sigma'(0) h_{ij} + \frac{\sigma''(0)}{2} h_{ij}^2 + \mathrm{H.O.T.}$$

mean

linear

This part (after centering) is also negligible.
operator norm $= \mathcal{O}_p(d^{-1/2})$

The more challenging regime

Truly nonlinear case:

$$A = \sigma(H) \qquad H = (h_{ij})_{i,j \leq n} \qquad h_{ij} = \mathcal{O}_p(1)$$

The more challenging regime

Truly nonlinear case:

$$A = \sigma(H) \qquad H = (h_{ij})_{i,j \leq n} \qquad h_{ij} = \mathcal{O}_p(1)$$

First studied for kernel random matrices: [Cheng & Singer, '13]

The more challenging regime

Truly nonlinear case:

$$A = \sigma(H) \qquad H = (h_{ij})_{i,j \leq n} \qquad h_{ij} = \mathcal{O}_p(1)$$

First studied for kernel random matrices: [Cheng & Singer, ’13]

Random neural networks: [Pennington & Worah, ’17], [Louart et al., ’18], [Mei & Montanari, ’19], [Goldt et al. ’19], [Gerace et al. ’19],

The more challenging regime

Truly nonlinear case:

$$A = \sigma(H) \qquad H = (h_{ij})_{i,j \leq n} \qquad h_{ij} = \mathcal{O}_p(1)$$

First studied for kernel random matrices: [Cheng & Singer, ’13]

Random neural networks: [Pennington & Worah, ’17], [Louart et al., ’18], [Mei & Montanari, ’19], [Goldt et al. ’19], [Gerace et al. ’19],

Equivalence principle:

Nonlinear model = Linear model + Noise

The more challenging regime

Truly nonlinear case:

$$A = \sigma(H) \qquad H = (h_{ij})_{i,j \leq n} \qquad h_{ij} = \mathcal{O}_p(1)$$

First studied for kernel random matrices: [Cheng & Singer, '13]

Random neural networks: [Pennington & Worah, '17], [Louart et al., '18], [Mei & Montanari, '19], [Goldt et al. '19], [Gerace et al. '19],

Equivalence principle:

Nonlinear model = Linear model + Noise

Nonlinear random matrix

$$A = \sigma(XX^\top)$$

The more challenging regime

Truly nonlinear case:

$$A = \sigma(H) \qquad H = (h_{ij})_{i,j \le n} \qquad h_{ij} = \mathcal{O}_p(1)$$

First studied for kernel random matrices: [Cheng & Singer, '13]

Random neural networks: [Pennington & Worah, '17], [Louart et al., '18], [Mei & Montanari, '19], [Goldt et al. '19], [Gerace et al. '19],

Equivalence principle:

Nonlinear model = Linear model + Noise

Nonlinear random matrix

$$A = \sigma(XX^\top)$$

$$\simeq$$

Linear model + (***Gaussian***) noise

$$B = \mu_0 1_{d\times d} + \mu_1 XX^\top + \mu_2^* Z$$

The more challenging regime

Truly nonlinear case:

$$A = \sigma(H) \qquad H = (h_{ij})_{i,j \le n} \qquad h_{ij} = \mathcal{O}_p(1)$$

First studied for kernel random matrices: [Cheng & Singer, '13]

Random neural networks: [Pennington & Worah, '17], [Louart et al., '18], [Mei & Montanari, '19], [Goldt et al. '19], [Gerace et al. '19],

Equivalence principle:

Nonlinear model = Linear model + Noise

Independent "noise"

Nonlinear random matrix

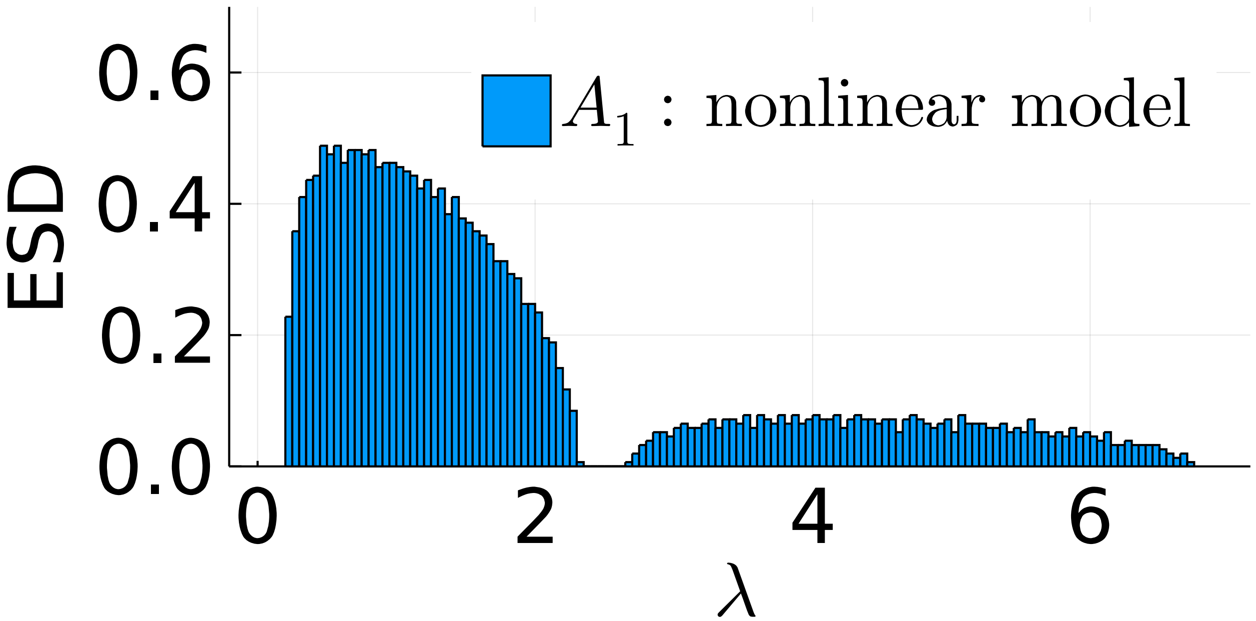
$$A = \sigma(XX^\top)$$

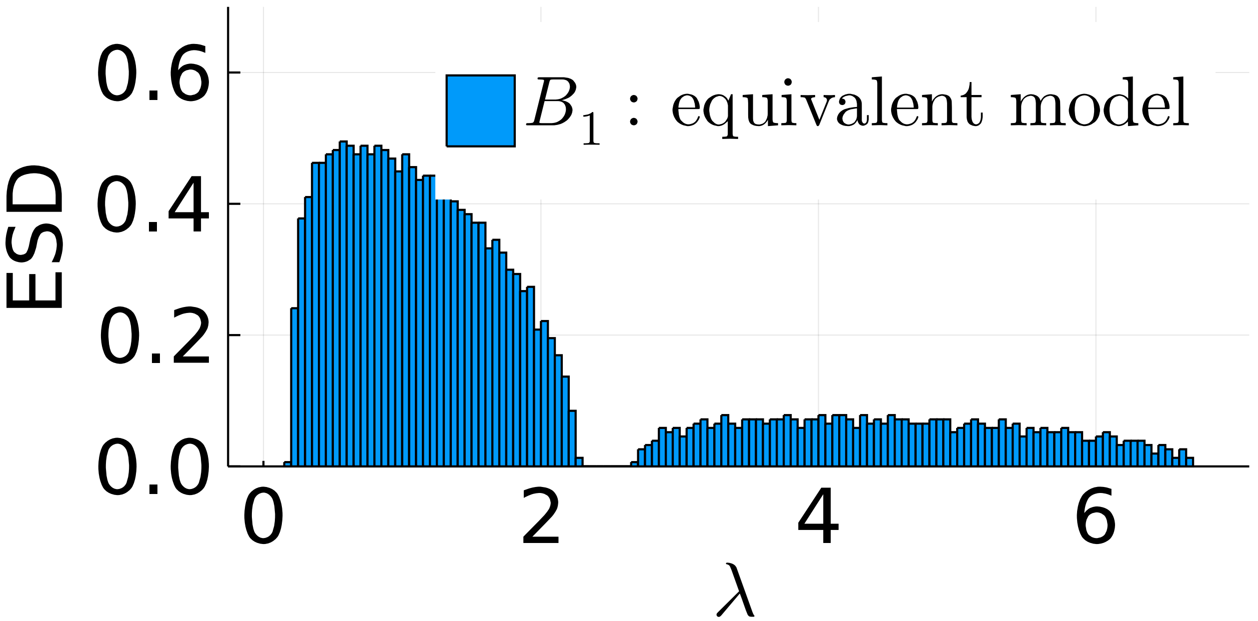
$$\simeq$$

Linear model + (***Gaussian***) noise

$$B = \mu_0 1_{d\times d} + \mu_1 XX^\top + \mu_2^* Z$$

Illustration of the equivalence principle





$$A_1 = \tanh(WX)$$

This lecture

- **Part I Random matrices:** kernel matrices, random features, and related models

Approximate rotational invariance of multilinear chaos

This lecture

- **Part I Random matrices:** kernel matrices, random features, and related models

  Approximate rotational invariance of multilinear chaos

- **Part II Beyond spectral equivalence: empirical risk minimization**

  Central limit theorems for Wiener chaos

This lecture

- **Part I** **Random matrices:** kernel matrices, random features, and related models

  Approximate rotational invariance of multilinear chaos

The staircase phenomenon in learning curves

Learning a function $f(x)$ defined on the hypersphere $\mathcal{S}^{d-1}$

Training set: $\{x_i, y_i = f(x_i)\}_{1 \le i \le n}$

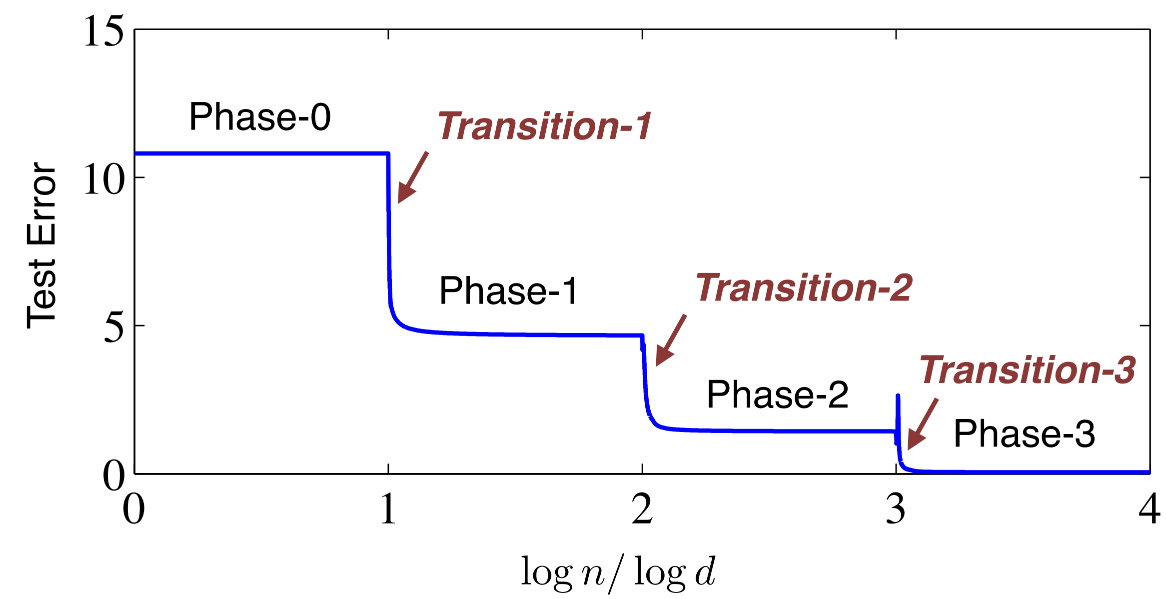
Algorithm: kernel ridge regression

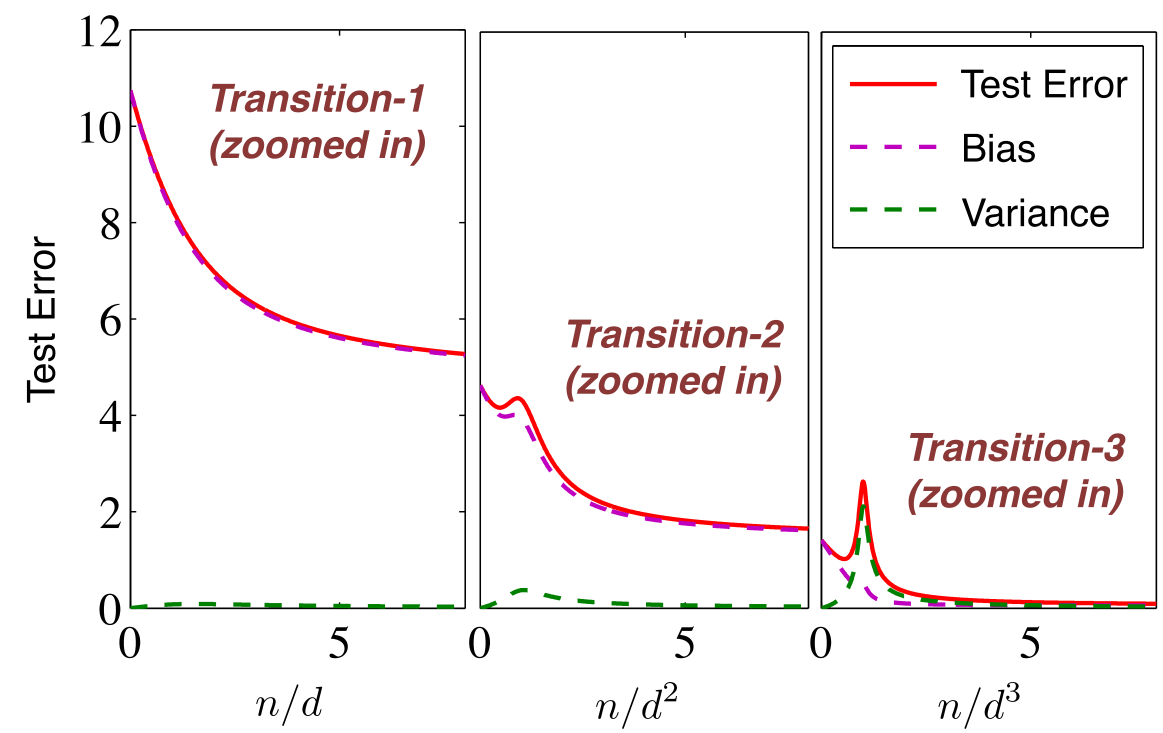
The staircase phenomenon in learning curves

Learning a function $f(x)$ defined on the hypersphere $\mathcal{S}^{d-1}$

Training set: $\{x_i, y_i = f(x_i)\}_{1 \le i \le n}$

Algorithm: kernel ridge regression





[Ghorbani et al., ’20], [Xiao et al. ’22], [Hu, Lu & Misiakiewicz ’24], …

The staircase phenomenon in learning curves

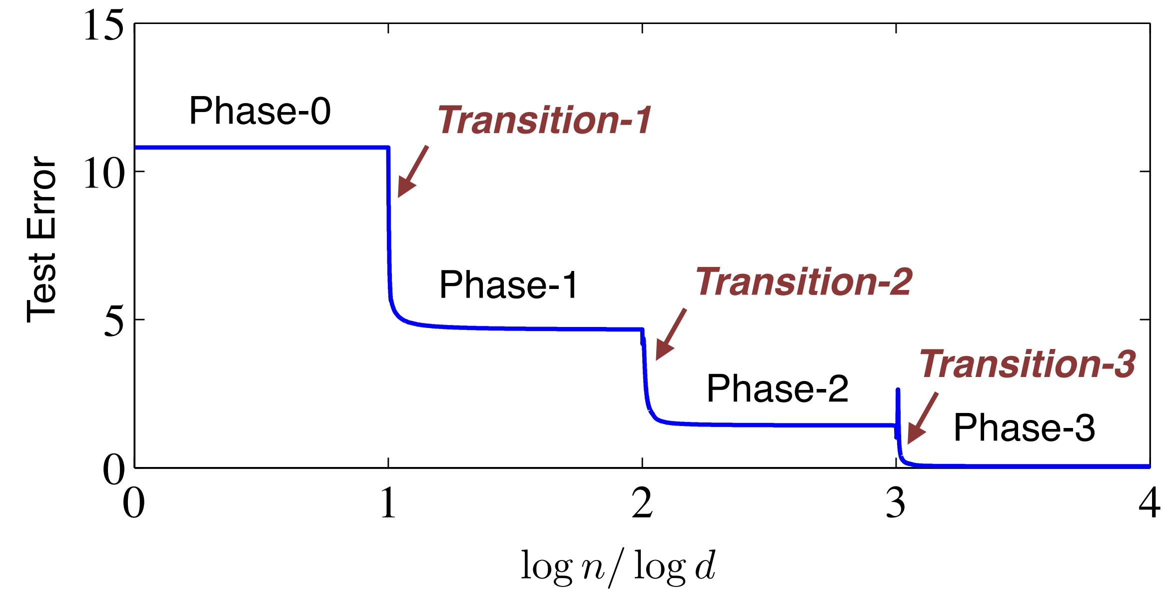
Learning a function $f(x)$ defined on the hypersphere $\mathcal{S}^{d-1}$

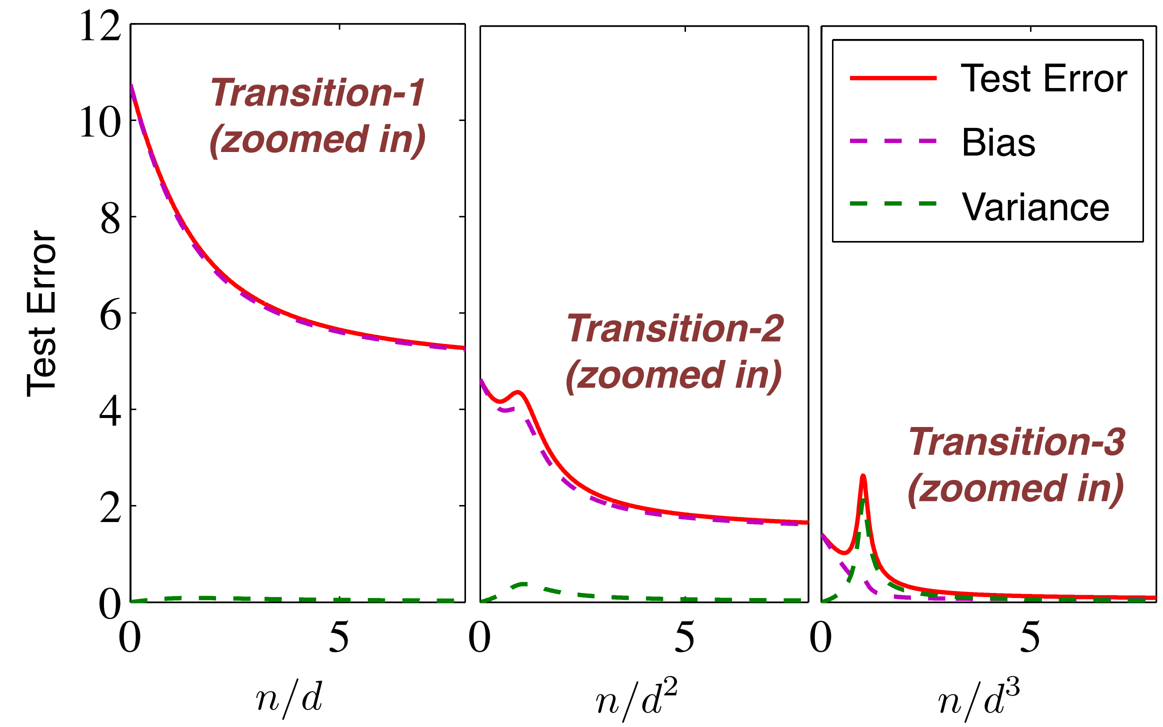
Training set: $\{x_i, y_i = f(x_i)\}_{1 \le i \le n}$

Algorithm: kernel ridge regression

Nonlinear random matrices in ***polynomial scaling regimes***:

$$\frac{n}{d^\ell} \to \alpha \quad \text{for } \ell = 1, 2, \ldots$$





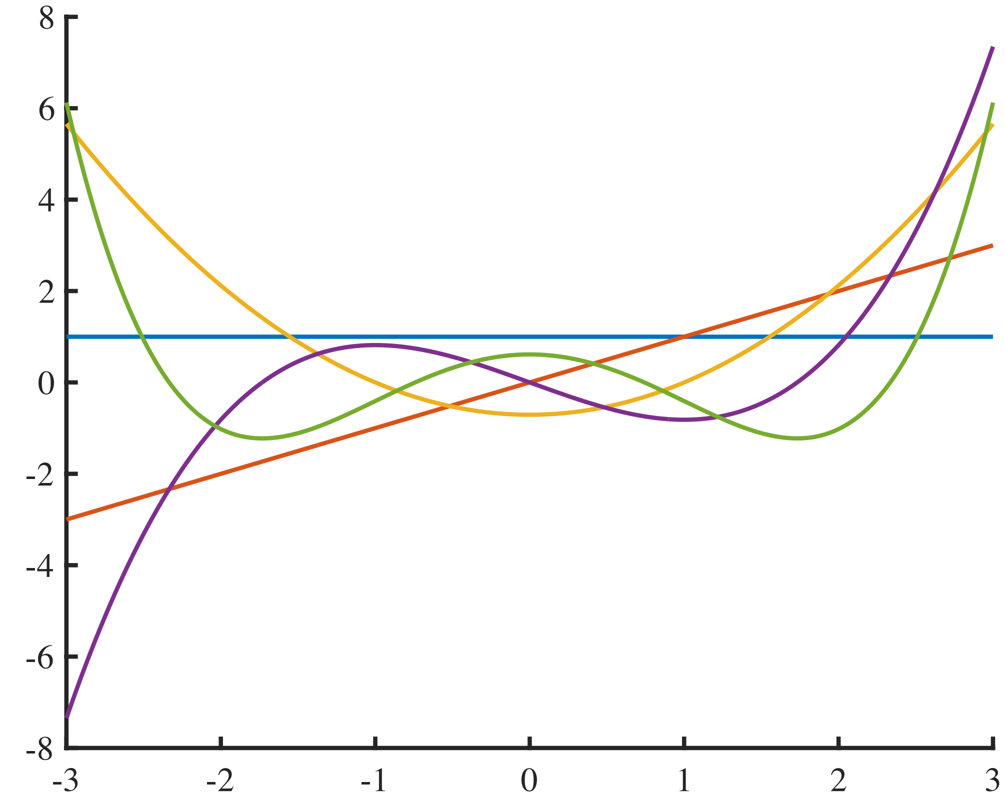
[Ghorbani et al., '20], [Xiao et al. '22], [Hu, Lu & Misiakiewicz '24], …

Kernel matrices beyond linear scaling regimes

Orthogonal polynomial expansion

Hermite polynomials: complete orthonormal basis

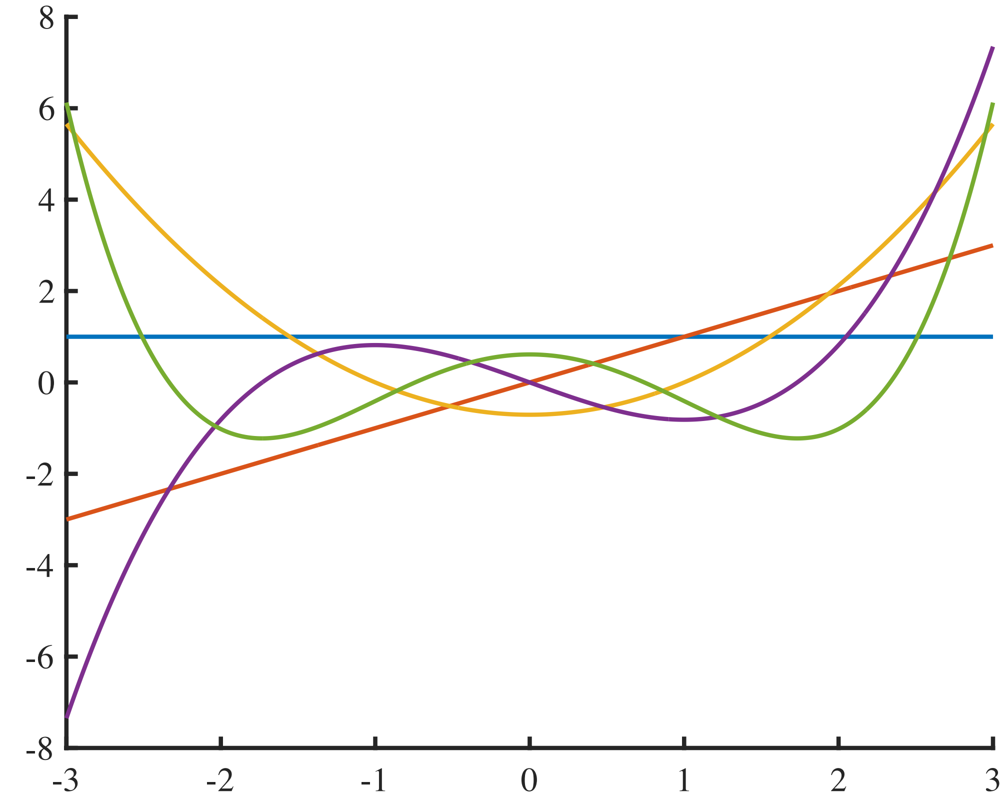
$$h_0(x) = 1$$
$$h_1(x) = x$$
$$h_2(x) = \frac{x^2 - 1}{\sqrt{2}}$$
$$h_3(x) = \frac{x^3 - 3x}{\sqrt{6}}$$



Orthogonal polynomial expansion

Hermite polynomials: complete orthonormal basis

$$h_0(x) = 1$$
$$h_1(x) = x$$
$$h_2(x) = \frac{x^2 - 1}{\sqrt{2}}$$
$$h_3(x) = \frac{x^3 - 3x}{\sqrt{6}}$$



Decomposition:

$$\sigma(x) = \mu_0 h_0(x) + \mu_1 h_1(x) + \mu_2 h_2(x) + \mu_3 h_3(x) + \dots$$

Inner product kernel random matrices

Given a collection of independent spherical vectors $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subset \mathbb{R}^d$:

$$A_{ij} = \begin{cases} \sigma(\boldsymbol{x}_i^\top \boldsymbol{x}_j), & i \neq j \\ 0, & i = j \end{cases}$$

Inner product kernel random matrices

Given a collection of independent spherical vectors $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subset \mathbb{R}^d$:

$$A_{ij} = \begin{cases} \sigma(\boldsymbol{x}_i^\top \boldsymbol{x}_j), & i \neq j \\ 0, & i = j \end{cases}$$

Decomposition:

$$\sigma(x) = \mu_0 h_0(x) + \mu_1 h_1(x) + \mu_2 h_2(x) + \mu_3 h_3(x) + \ldots$$

and thus

$$\boxed{\boldsymbol{A} = \mu_0 \boldsymbol{H}_0 + \mu_1 \boldsymbol{H}_1 + \mu_2 \boldsymbol{H}_2 + \mu_3 \boldsymbol{H}_3 + \ldots}$$

where

$$(H_k)_{ij} = \begin{cases} h_k(\boldsymbol{x}_i^\top \boldsymbol{x}_j), & i \neq j \\ 0, & i = j \end{cases}$$

Kernel random matrix beyond the linear scaling regime

[Lu and Yau, arXiv:2205.06308]

Equivalence phenomenon: $n = \alpha d^{\ell}$ for some $\alpha > 0$ and $\ell \in \mathbb{N}$:

$$\boldsymbol{A} = \mu_0 \boldsymbol{H}_0 + \ldots + \mu_{\ell-1} \boldsymbol{H}_{\ell-1} + \mu_\ell \boldsymbol{H}_\ell + \mu_{\ell+1} \boldsymbol{H}_{\ell+1} + \mu_{\ell+2} \boldsymbol{H}_{\ell+2} + \ldots$$

Kernel random matrix beyond the linear scaling regime

[Lu and Yau, arXiv:2205.06308]

Equivalence phenomenon: $n = \alpha d^\ell$ for some $\alpha > 0$ and $\ell \in \mathbb{N}$:

$$\boldsymbol{A} = \underbrace{\mu_0 \boldsymbol{H}_0 + \ldots + \mu_{\ell-1} \boldsymbol{H}_{\ell-1}}_{\text{Low-rank components}} + \mu_\ell \boldsymbol{H}_\ell + \mu_{\ell+1} \boldsymbol{H}_{\ell+1} + \mu_{\ell+2} \boldsymbol{H}_{\ell+2} + \ldots$$

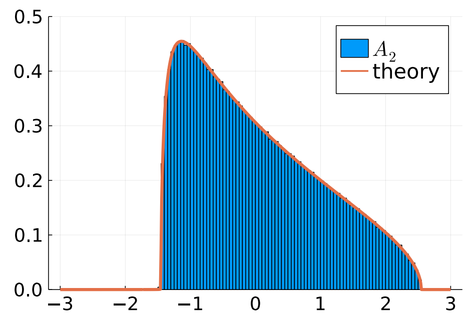
Kernel random matrix beyond the linear scaling regime

[Lu and Yau, arXiv:2205.06308]

Equivalence phenomenon: $n = \alpha d^{\ell}$ for some $\alpha > 0$ and $\ell \in \mathbb{N}$:

$$\boldsymbol{A} = \underbrace{\mu_0 \boldsymbol{H}_0 + \ldots + \mu_{\ell-1} \boldsymbol{H}_{\ell-1}}_{\text{Low-rank components}} + \mu_{\ell} \boldsymbol{H}_{\ell} + \mu_{\ell+1} \boldsymbol{H}_{\ell+1} + \mu_{\ell+2} \boldsymbol{H}_{\ell+2} + \ldots$$

Marchenko-Pastur law



Kernel random matrix beyond the linear scaling regime

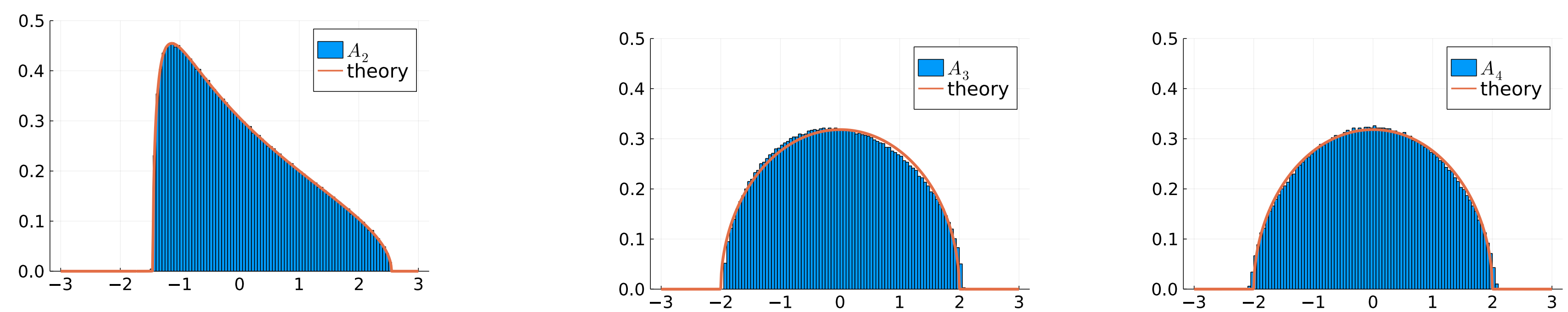
[Lu and Yau, arXiv:2205.06308]

Equivalence phenomenon: $n = \alpha d^{\ell}$ for some $\alpha > 0$ and $\ell \in \mathbb{N}$:

$$\boldsymbol{A} = \underbrace{\mu_0 \boldsymbol{H}_0 + \ldots + \mu_{\ell-1} \boldsymbol{H}_{\ell-1}}_{\text{Low-rank components}} + \mu_\ell \boldsymbol{H}_\ell + \mu_{\ell+1} \boldsymbol{H}_{\ell+1} + \mu_{\ell+2} \boldsymbol{H}_{\ell+2} + \ldots$$

Marchenko-Pastur law

Independent GOE matrices (noise)



Kernel random matrix beyond the linear scaling regime

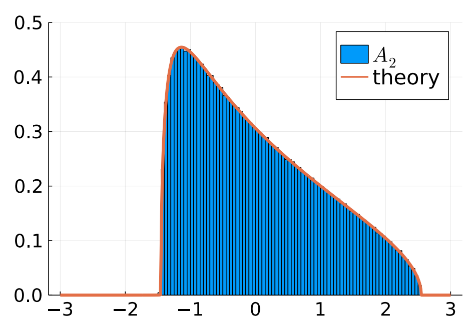
[Lu and Yau, arXiv:2205.06308]

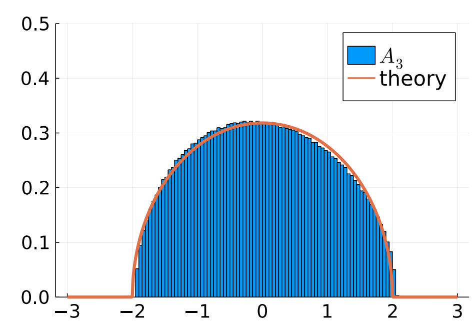
Equivalence phenomenon: $n = \alpha d^{\ell}$ for some $\alpha > 0$ and $\ell \in \mathbb{N}$:

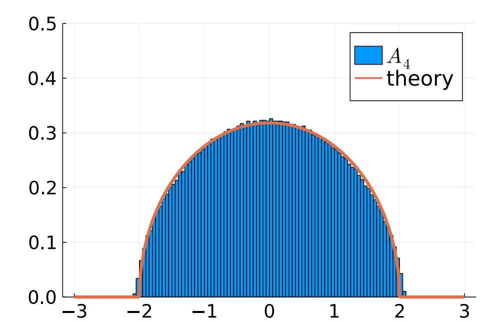
$$\boldsymbol{A} = \underbrace{\mu_0 \boldsymbol{H}_0 + \ldots + \mu_{\ell-1} \boldsymbol{H}_{\ell-1}}_{\text{Low-rank components}} + \mu_{\ell} \boldsymbol{H}_{\ell} + \mu_{\ell+1} \boldsymbol{H}_{\ell+1} + \mu_{\ell+2} \boldsymbol{H}_{\ell+2} + \ldots$$

Marchenko-Pastur law

Independent GOE matrices
(noise)







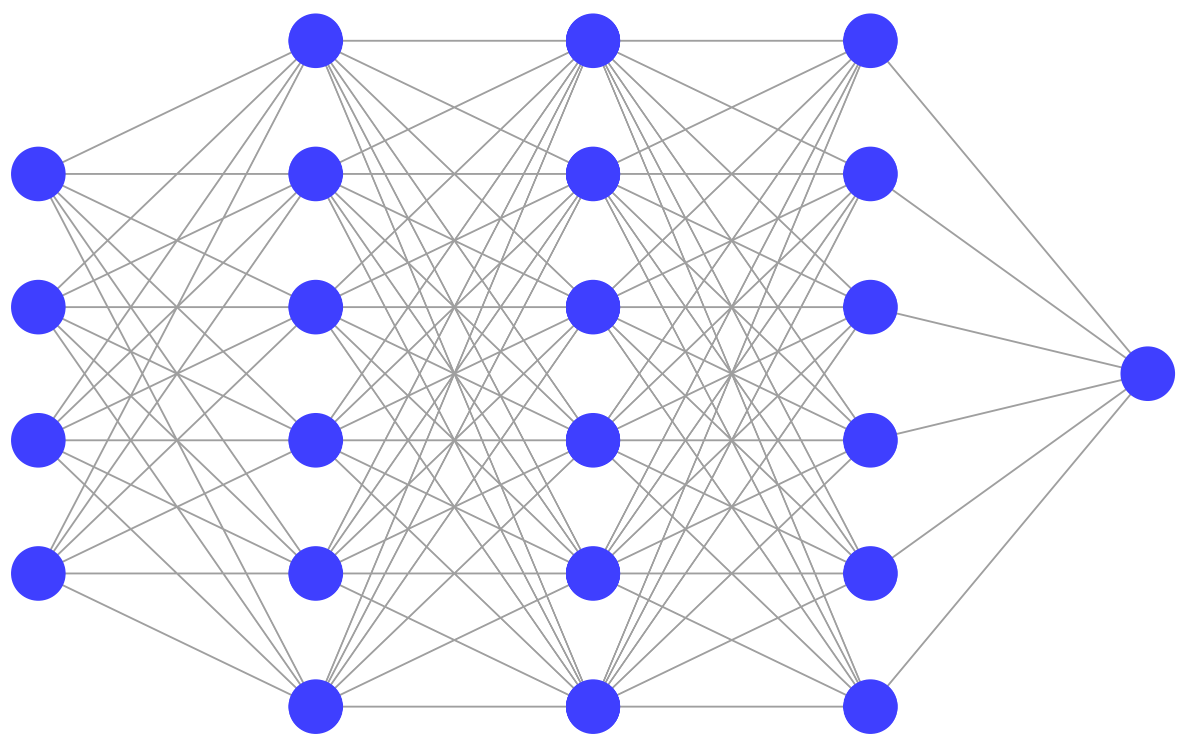
Generalizes [Cheng & Singer, ’13], special case of $\ell = 1$ (linear scaling)

Dubova, Lu, McKenna, Yau, arXiv:2310.18280 (universality)

Related models

Random feature regression beyond linear scalings

Input:
$x \in \mathbb{R}^d$



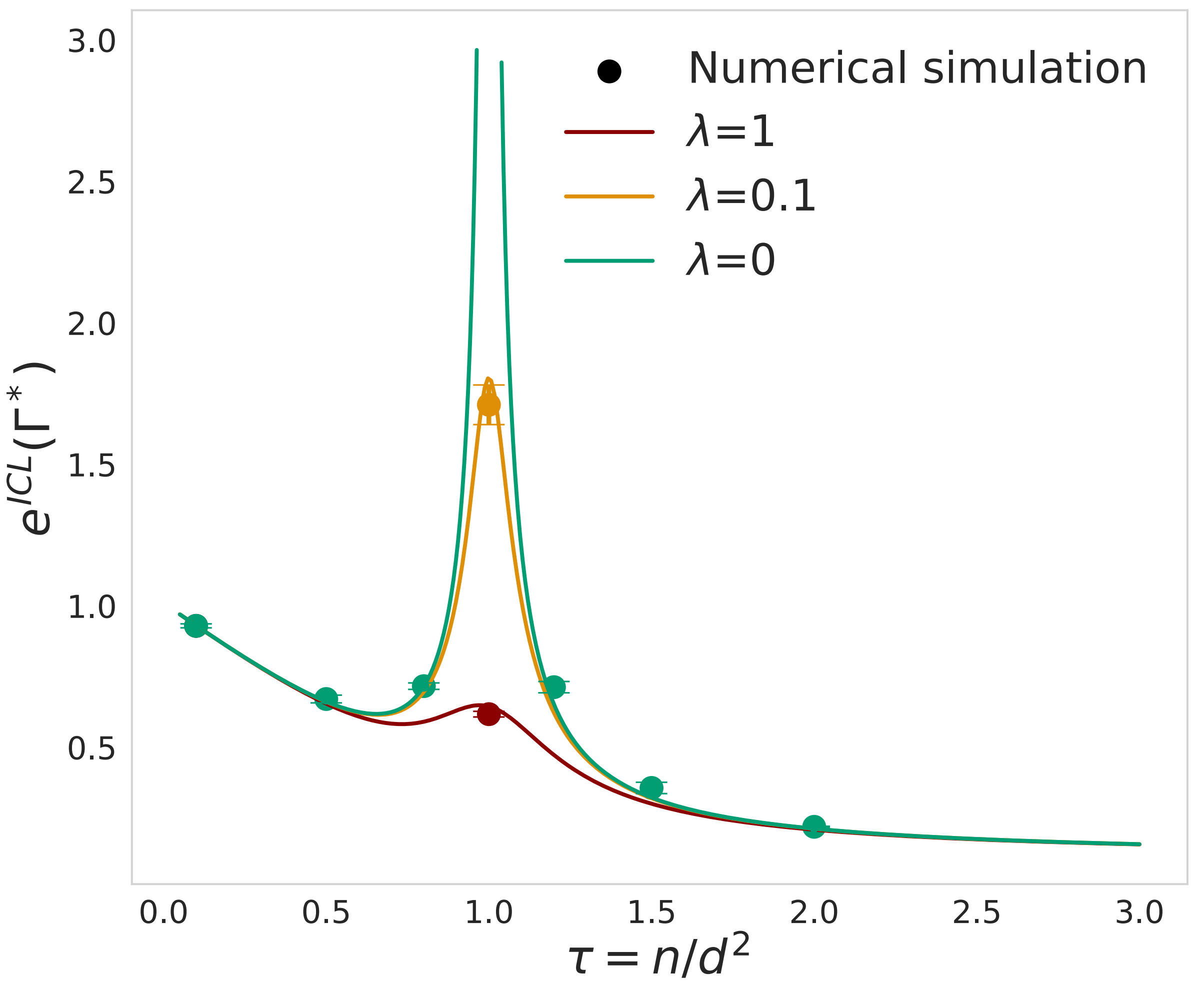
Output:
$\hat{y} = \langle f, w \rangle$

Feature: $f = \sigma(W_3\sigma(W_2\sigma(W_1 x)))$

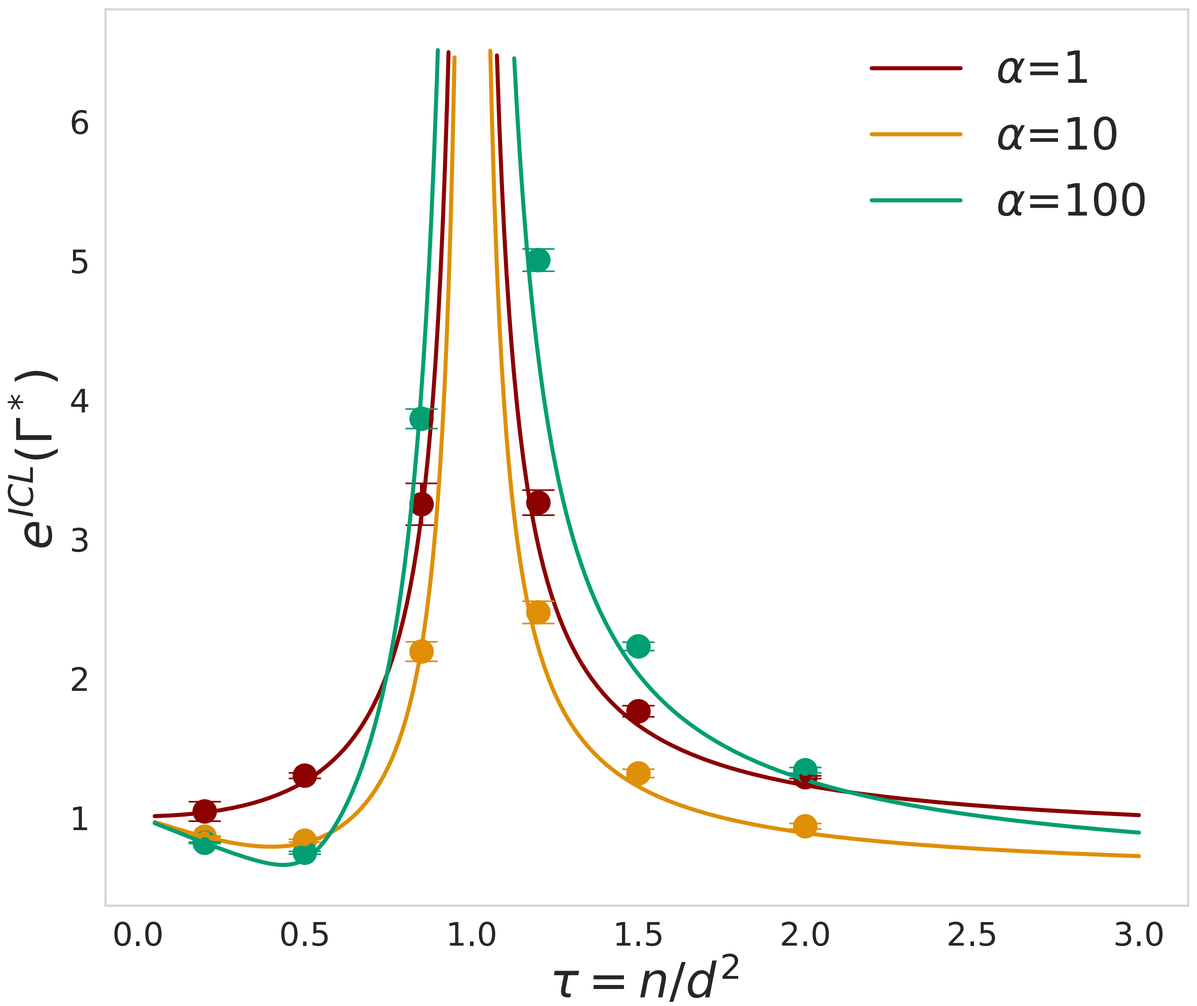
$\sigma(\cdot)$: ***activation function***

Hu, Lu & Misiakiewicz, arXiv:2403.08160

In-context learning using linear attention



ICL learning curve



ICL learning curve (ridgeless)

Lu, Letey, Zavatone-Veth, Maiti & Pehlevan,
"Asymptotic theory of in-context learning by linear attention,"
Proceedings of the National Academy of Sciences (PNAS), 2025.

Other related models

- Hadamard product of independent sample covariance matrices

$$(X^\top X) \odot (Y^\top Y)$$

Other related models

- Hadamard product of independent sample covariance matrices

$$(X^\top X) \odot (Y^\top Y) \quad \simeq \quad \text{Marchenko-Pastur Law}$$

[Assaly and Benigni ’25]

Other related models

- Hadamard product of independent sample covariance matrices

$$(X^\top X) \odot (Y^\top Y) \quad \simeq \quad \text{Marchenko-Pastur Law}$$

[Assaly and Benigni '25]

- NTK kernel matrices

$$K = (X^\top X) \odot \left[\sigma'(X^\top W^\top)\mathrm{diag}(a_1, \ldots, a_p)\sigma'(WX)\right] + \sigma(X^\top W^\top)\sigma(WX)$$

[Benigni and Paquette '25]

This lecture

- **Part I Random matrices:** kernel matrices, random features, and related models

  Approximate rotational invariance of polynomial chaos

- **Part II Beyond spectral equivalence: empirical risk minimization**

  Central limit theorems for Wiener chaos

This lecture

- **Part II** **Beyond spectral equivalence: empirical risk minimization**

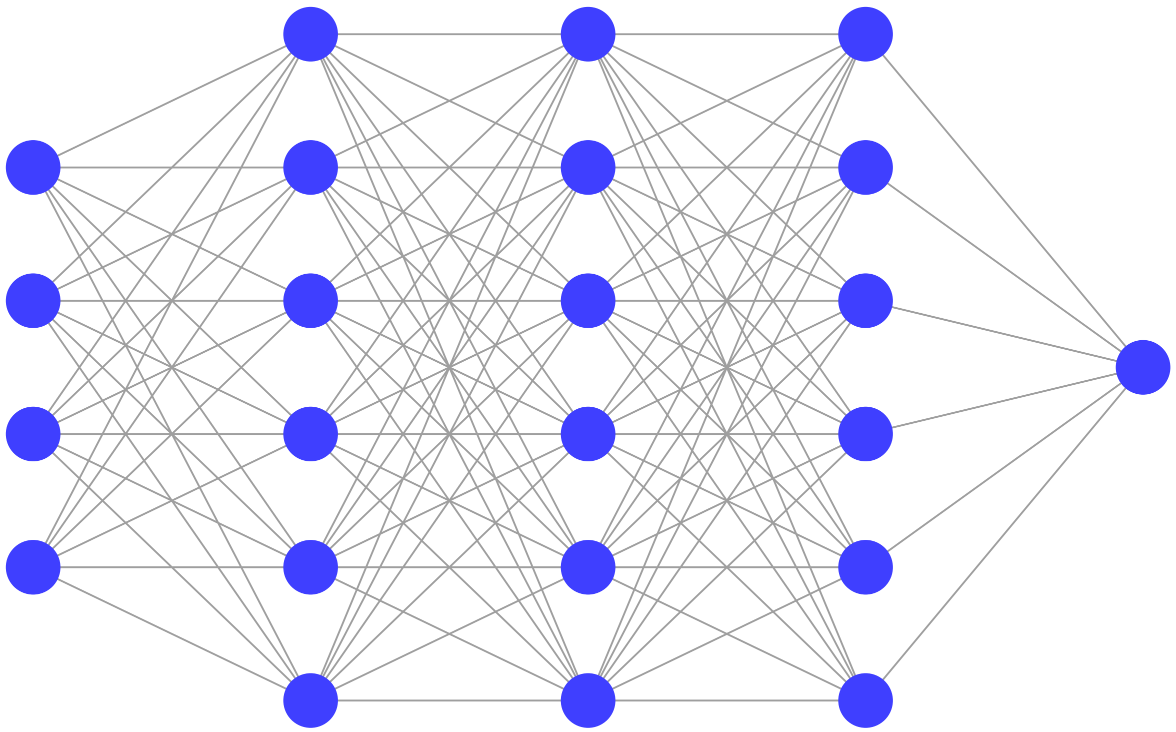
  Central limit theorems for Wiener chaos

Multilayer perceptrons

Feature: $f = \sigma(W_3\sigma(W_2\sigma(W_1x)))$

$\sigma(\cdot)$: ***activation function***

Input:
$x \in \mathbb{R}^d$



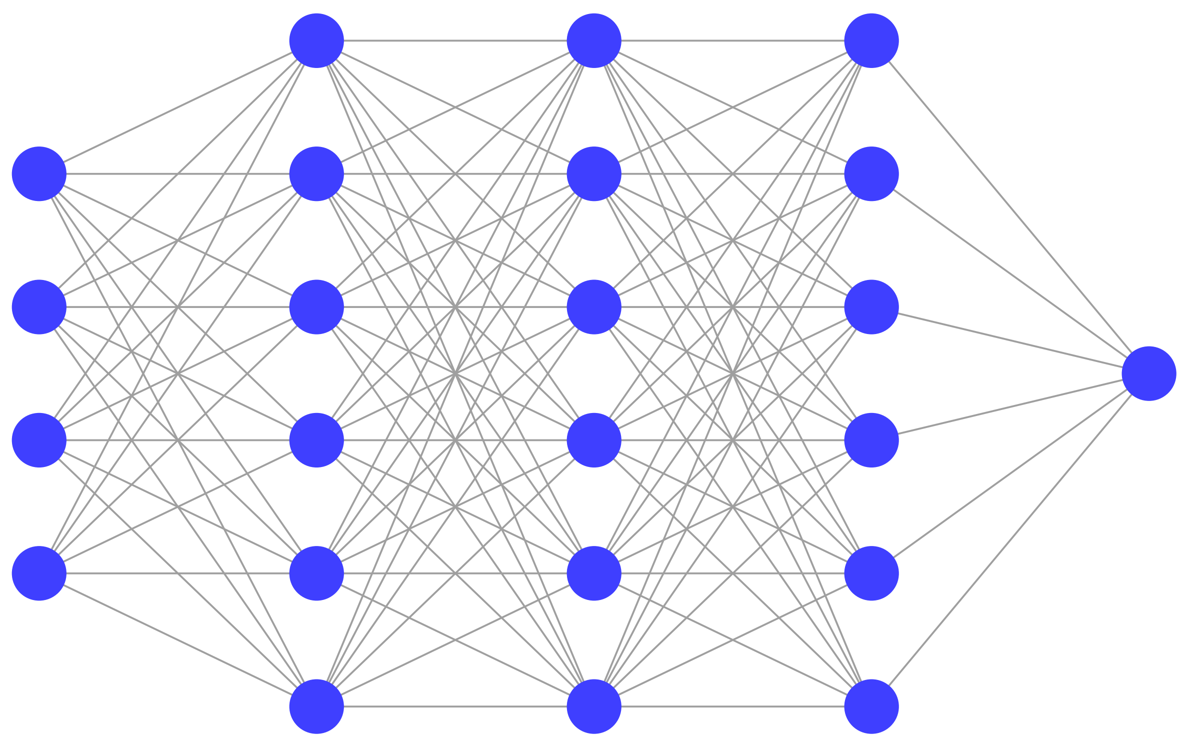
Output:
$\hat{y} = \langle f, w \rangle$

Multilayer perceptrons

Feature: $f = \sigma(W_3\sigma(W_2\sigma(W_1x)))$

$\sigma(\cdot)$: ***activation function***

Input:
$x \in \mathbb{R}^d$



Output:
$\hat{y} = \langle f, w \rangle$

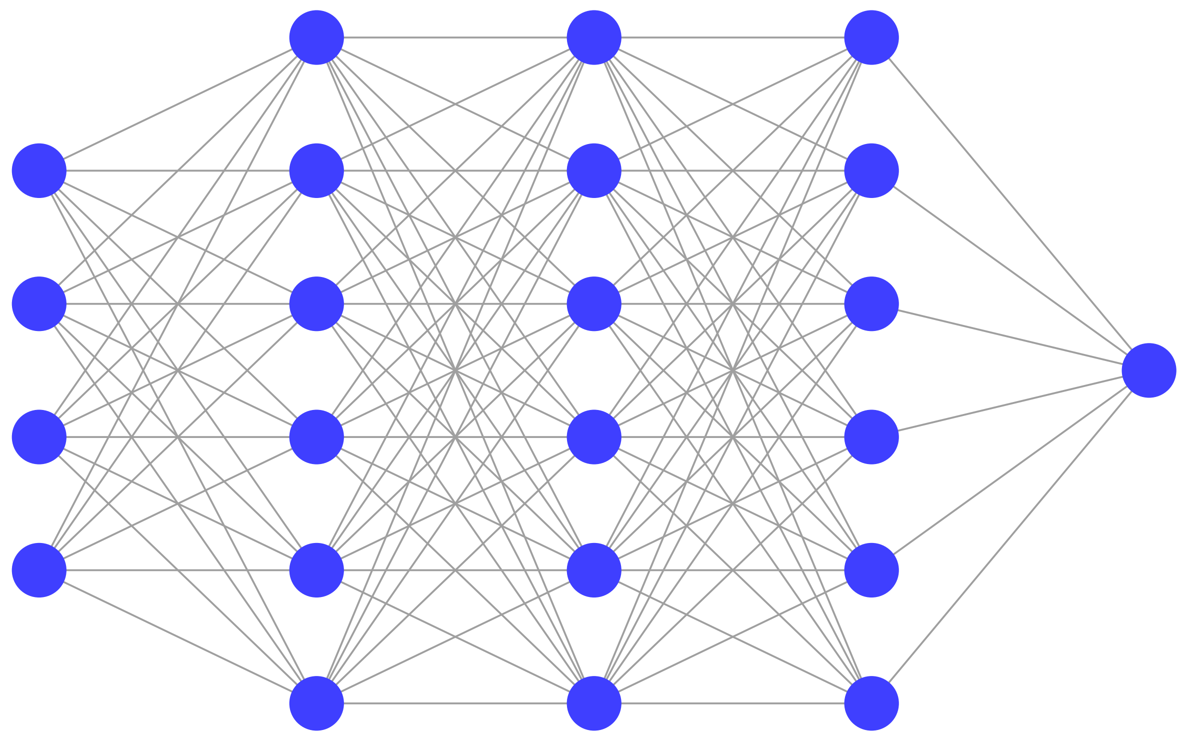
Given training data $\{x_i, y_i\}_{1\leq i\leq n}$, learning matrices W_1, W_2, W_3 and vector w

$$\min_{W_1, W_2, W_3, w} \sum_{i\leq n} \ell(\hat{y}_i, y_i)$$

The random feature model

Feature: $f = \sigma(W_3\sigma(W_2\sigma(W_1x)))$

$\sigma(\cdot)$: ***activation function***



Input:
$x \in \mathbb{R}^d$

Output:
$\hat{y} = \langle f, w \rangle$

Random feature model: [Rahimi & Recht, '08]

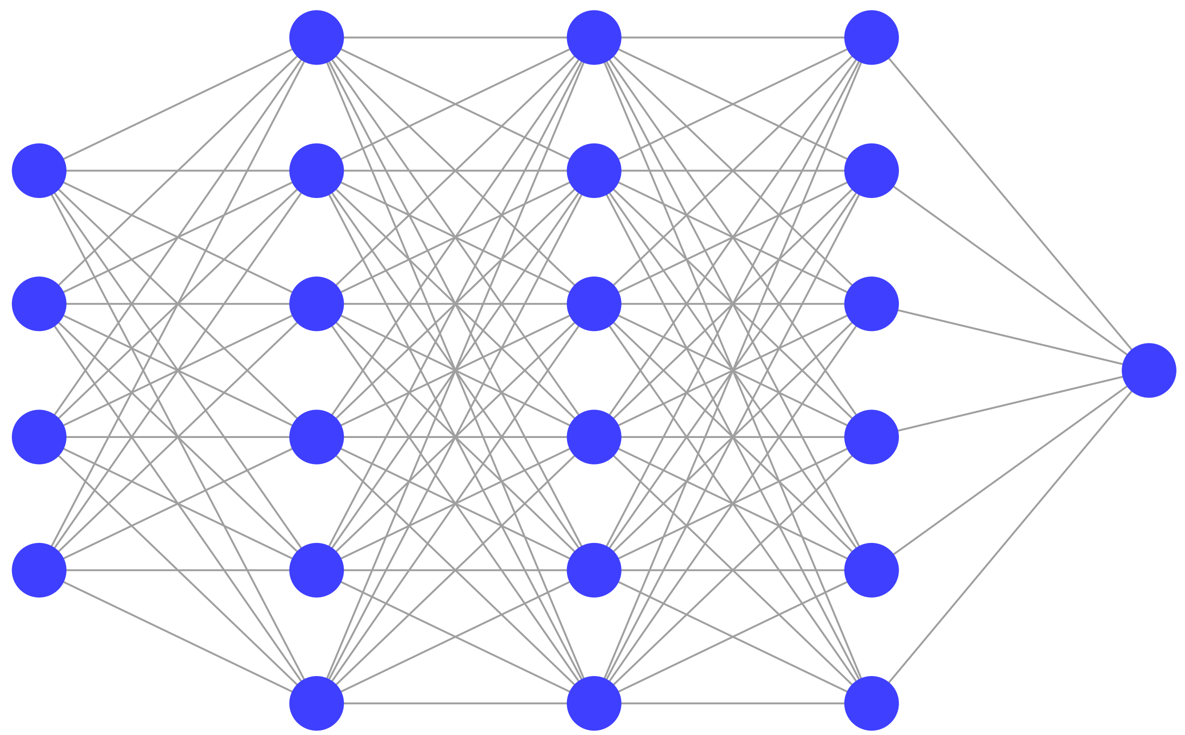
- W_1, W_2, W_3 : weight matrices are ***randomly initialized*** and then ***"frozen"***

The random feature model

Feature: $f = \sigma(W_3\sigma(W_2\sigma(W_1 x)))$

$\sigma(\cdot)$: ***activation function***

Input:
$x \in \mathbb{R}^d$



Output:
$\hat{y} = \langle f, w \rangle$

Random feature model: [Rahimi & Recht, '08]

- W_1, W_2, W_3 : weight matrices are ***randomly initialized*** and then ***"frozen"***

Only learn the ***weight vector*** in the last layer:

$$\arg\min_{w} \sum_{i\leq n} \ell(w^\top \sigma(W_3\sigma(W_2\sigma(W_1 x_i))); y_i)$$

Theoretical analysis in high-dimensions

Empirical risk minimization

$$w^* = \arg\min_w \sum_{i \leq n} \ell(w^\top f_i; y_i)$$

Feature vectors: $f_i = \sigma(W_3 \sigma(W_2 \sigma(W_1 x_i)))$

Theoretical analysis in high-dimensions

Empirical risk minimization

$$w^* = \arg\min_w \textstyle\sum_{i \le n} \ell(w^\top f_i; y_i)$$

Feature vectors: $f_i = \sigma(W_3 \sigma(W_2 \sigma(W_1 x_i)))$

Convex optimization problem involving a nonlinear random matrix

$$\boxed{F = [f_1, \ldots, f_n] = \sigma(W_3 \sigma(W_2 \sigma(W_1 X))) \in \mathbb{R}^{p \times n}}$$

Theoretical analysis in high-dimensions

Empirical risk minimization

$$w^* = \arg\min_w \sum_{i\leq n} \ell(w^\top f_i; y_i)$$

Feature vectors: $f_i = \sigma(W_3\sigma(W_2\sigma(W_1 x_i)))$

Convex optimization problem involving a nonlinear random matrix

$$\boxed{F = [f_1, \ldots, f_n] = \sigma(W_3\sigma(W_2\sigma(W_1 X))) \in \mathbb{R}^{p\times n}}$$

Goal: characterize the high-dimensional limits ($p, d, n \to \infty$) of

Train error:

$$\mathcal{E}_{\text{train}} = \min_w \sum_{i\leq n} \ell(w^\top f_i; y_i)$$

Test error:

$$\mathcal{E}_{\text{test}} = \mathbb{E}\big[\ell((w^*)^\top f_{\text{new}}; y_{\text{new}})\big]$$

Theoretical analysis in high-dimensions

Special case: linear activation function $\sigma(x) = x$

$$F = \sigma(W_3\sigma(W_2\sigma(W_1X))) \longrightarrow F = W_3W_2W_1X$$

Theoretical analysis in high-dimensions

Special case: linear activation function $\sigma(x) = x$

$$F = \sigma(W_3\sigma(W_2\sigma(W_1X))) \longrightarrow F = W_3W_2W_1X$$

Many powerful tools developed in the past decade:

$$w^* = \arg\min_w \sum_{i \leq n} \ell(w^\top f_i; y_i)$$

- Convex Gaussian minimax theorem (CGMT)
- Gaussian width, statistical dimensions
- Approximate message passing (AMP) and variants

Theoretical analysis in high-dimensions

Special case: linear activation function $\sigma(x) = x$

$$F = \sigma(W_3\sigma(W_2\sigma(W_1X))) \longrightarrow F = W_3W_2W_1X$$

Many powerful tools developed in the past decade:

$$w^* = \arg\min_w \sum_{i \leq n} \ell(w^\top f_i; y_i)$$

- Convex Gaussian minimax theorem (CGMT)
- Gaussian width, statistical dimensions
- Approximate message passing (AMP) and variants

Challenge: nonlinear activation function breaks the standard statistical assumptions of these analysis techniques

Two versions of the feature maps

***Nonlinear* feature map:**

$$A = \sigma(WX)$$

Two versions of the feature maps

***Nonlinear* feature map:**

$$A = \sigma(WX)$$

***Noisy linear* feature map:**

$$B = \mu_0 1_{p \times n} + \mu_1 WX + \mu_2 Z$$

where $z_{ij} \sim_{\text{iid}} \mathcal{N}(0, 1)$

Two versions of the feature maps

Nonlinear feature map:

$$A = \sigma(WX)$$

Noisy linear feature map:

$$B = \mu_0 1_{p \times n} + \mu_1 WX + \mu_2 Z$$

where $z_{ij} \sim_{\mathrm{iid}} \mathcal{N}(0,1)$

$\mu_0 = \mathbb{E}[\sigma(g)] \quad g \sim \mathcal{N}(0,1)$
$\mu_1 = \mathbb{E}[G\sigma(g)]$
$\mu_2 = (\mathbb{E}[\sigma^2(g)] - \mu_0^2 - \mu_1^2)^{1/2}$

Two versions of the feature maps

Nonlinear feature map:

$$A = \sigma(WX)$$

Training error:

$$\mathcal{E}_{\text{train}}(A) = \min_{w} \sum_{i\le n} \ell(a_i^\top w; y_i)$$

Noisy linear feature map:

$$B = \mu_0 1_{p\times n} + \mu_1 WX + \mu_2 Z$$

where $z_{ij} \sim_{\text{iid}} \mathcal{N}(0,1)$

$\mu_0 = \mathbb{E}[\sigma(g)] \quad g \sim \mathcal{N}(0,1)$
$\mu_1 = \mathbb{E}[G\sigma(g)]$
$\mu_2 = (\mathbb{E}[\sigma^2(g)] - \mu_0^2 - \mu_1^2)^{1/2}$

$$\mathcal{E}_{\text{train}}(B) = \min_{w} \sum_{i\le n} \ell(b_i^\top w; y_i)$$

Two versions of the feature maps

***Nonlinear** feature map:*

$$A = \sigma(WX)$$

***Noisy linear** feature map:*

$$B = \mu_0 1_{p \times n} + \mu_1 WX + \mu_2 Z$$

where $z_{ij} \sim_{\text{iid}} \mathcal{N}(0,1)$

$$\mu_0 = \mathbb{E}[\sigma(g)] \quad g \sim \mathcal{N}(0,1)$$
$$\mu_1 = \mathbb{E}[G\sigma(g)]$$
$$\mu_2 = (\mathbb{E}[\sigma^2(g)] - \mu_0^2 - \mu_1^2)^{1/2}$$

Training error:

$$\mathcal{E}_{\text{train}}(A) = \min_w \sum_{i \le n} \ell(a_i^\top w; y_i)$$

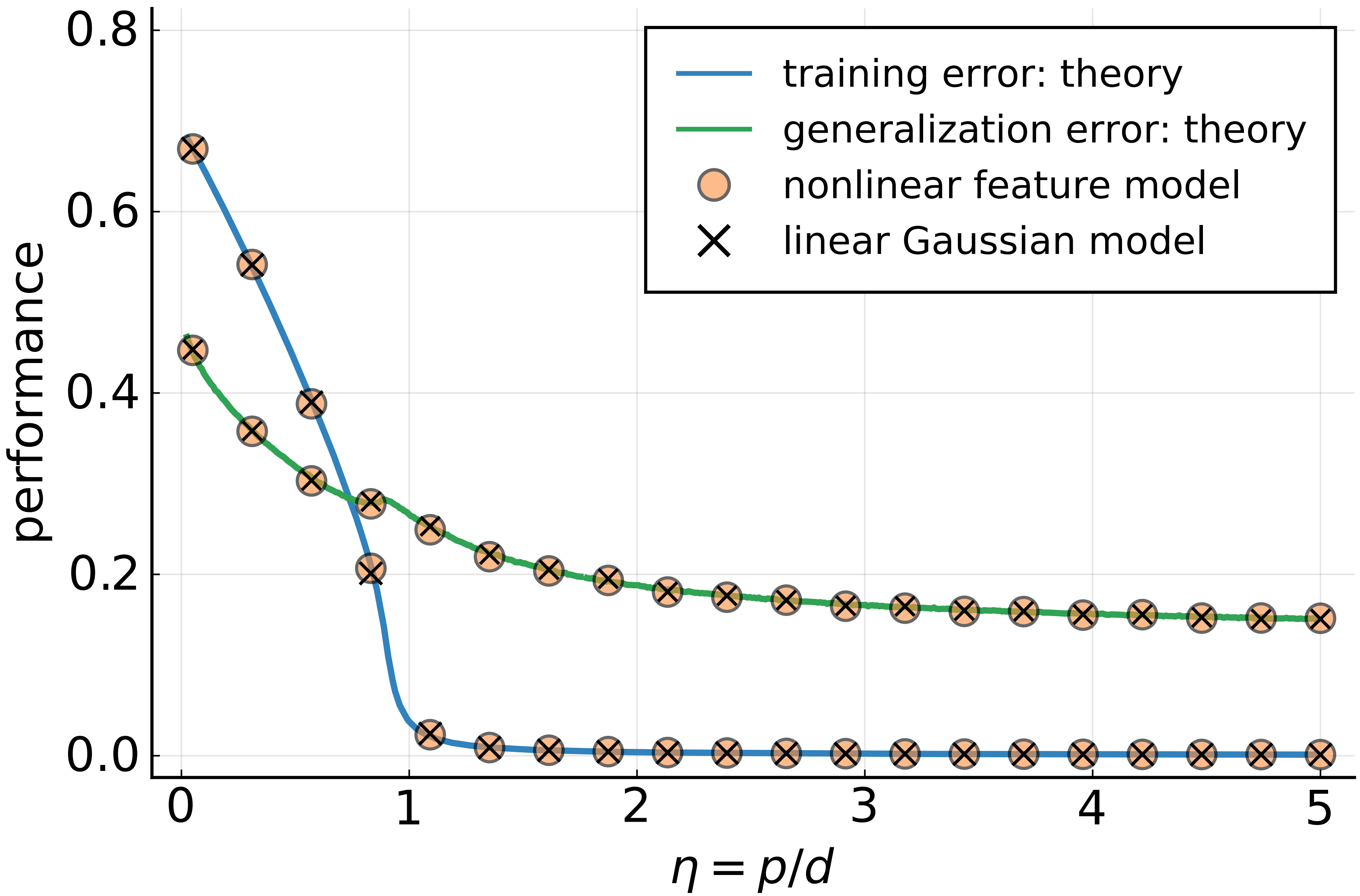
$$\mathcal{E}_{\text{train}}(B) = \min_w \sum_{i \le n} \ell(b_i^\top w; y_i)$$

Test error:

$$\mathcal{E}_{\text{test}}(A) = \mathbb{E}\big[\ell(a_{\text{new}}^\top w_A^*; y_{\text{new}})\big]$$

$$\mathcal{E}_{\text{test}}(B) = \mathbb{E}\big[\ell(b_{\text{new}}^\top w_B^*; y_{\text{new}})\big]$$

Numerical surprises



Logistic regression $\sigma(x) = \tanh(x)$

Gaussian equivalence phenomenon

***Nonlinear* feature map:**

$$A = \sigma(WX)$$

***Noisy linear* feature map:**

$$B = \mu_0 1_{p\times n} + \mu_1 WX + \mu_2 Z$$

where $z_{ij} \sim_{\text{iid}} \mathcal{N}(0,1)$

Gaussian equivalence phenomenon

***Nonlinear* feature map:**

$$A = \sigma(WX)$$

***Noisy linear* feature map:**

$$B = \mu_0 1_{p \times n} + \mu_1 WX + \mu_2 Z$$

where $z_{ij} \sim_{\text{iid}} \mathcal{N}(0,1)$

Gaussian equivalence:

For all "*generic*" W and "*reasonable*" $\sigma(\cdot)$,

$$\lim_{p\to\infty} \frac{1}{p}\mathcal{E}_{\text{train}}(A) = \lim_{p\to\infty} \frac{1}{p}\mathcal{E}_{\text{train}}(B) \quad \text{and} \quad \lim_{p\to\infty} \frac{1}{p}\mathcal{E}_{\text{test}}(A) = \lim_{p\to\infty} \frac{1}{p}\mathcal{E}_{\text{test}}(B)$$

Stated as a conjecture in:

[Goldt et al. '19, '20], [Mei & Montanari, '19]

Exploited in: [Montanari, Ruan, Sohn, Yan, '19], [Oussama & Lu, '20] …

Why is it useful?

Exploiting the Gaussian equivalence

Sharp asymptotic analysis by exploiting the Gaussian equivalence:

[Dhifallah & Lu, arXiv:2008.11904]: Convex Gaussian minmax theorem (CGMT) for correlated feature vectors

Exploiting the Gaussian equivalence

Sharp asymptotic analysis by exploiting the Gaussian equivalence:

[Dhifallah & Lu, arXiv:2008.11904]: Convex Gaussian minmax theorem (CGMT) for correlated feature vectors

Theorem: [Dhifallah & Lu, '20] (informal) Convex loss functions and regularizers:

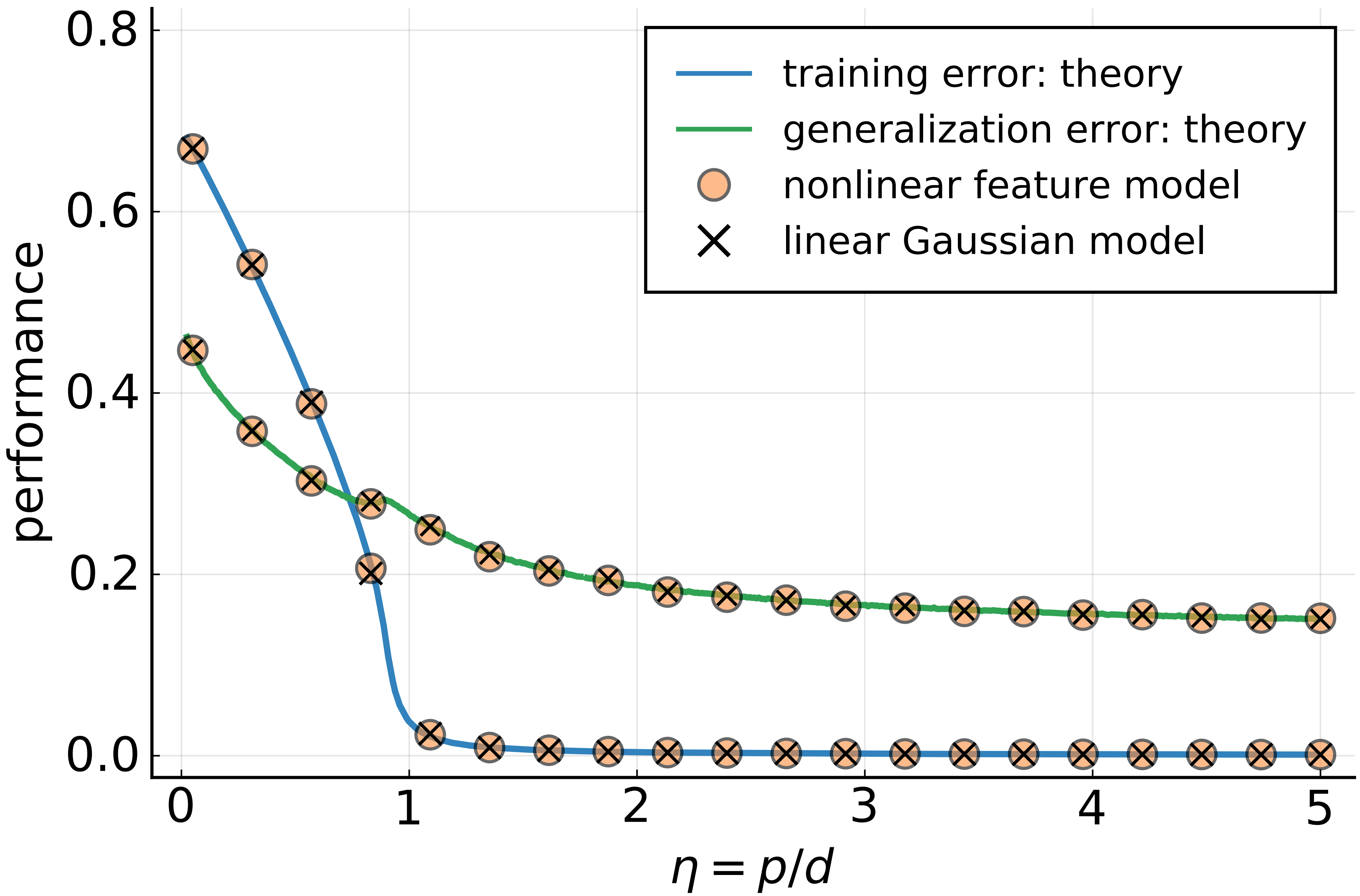
$$\mathcal{E}_{\text{train}}(\boldsymbol{B}) \xrightarrow{d,n\to\infty} C(t^*, \tau^*) \qquad \mathcal{E}_{\text{test}}(\boldsymbol{B}) \xrightarrow{d,n\to\infty} \mathbf{E} f(\nu_1, \nu_2)$$

$$\nu_1, \nu_2 \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}^*)$$

where the parameters t^*, τ^* and $\mathbf{\Sigma}^*$ are determined by some fixed point equations

Related work that also exploits this conjecture: [Gerace et al. '19], [Goldt et al. '19], [Montanari et al., '19], [Bosch et al., '22]

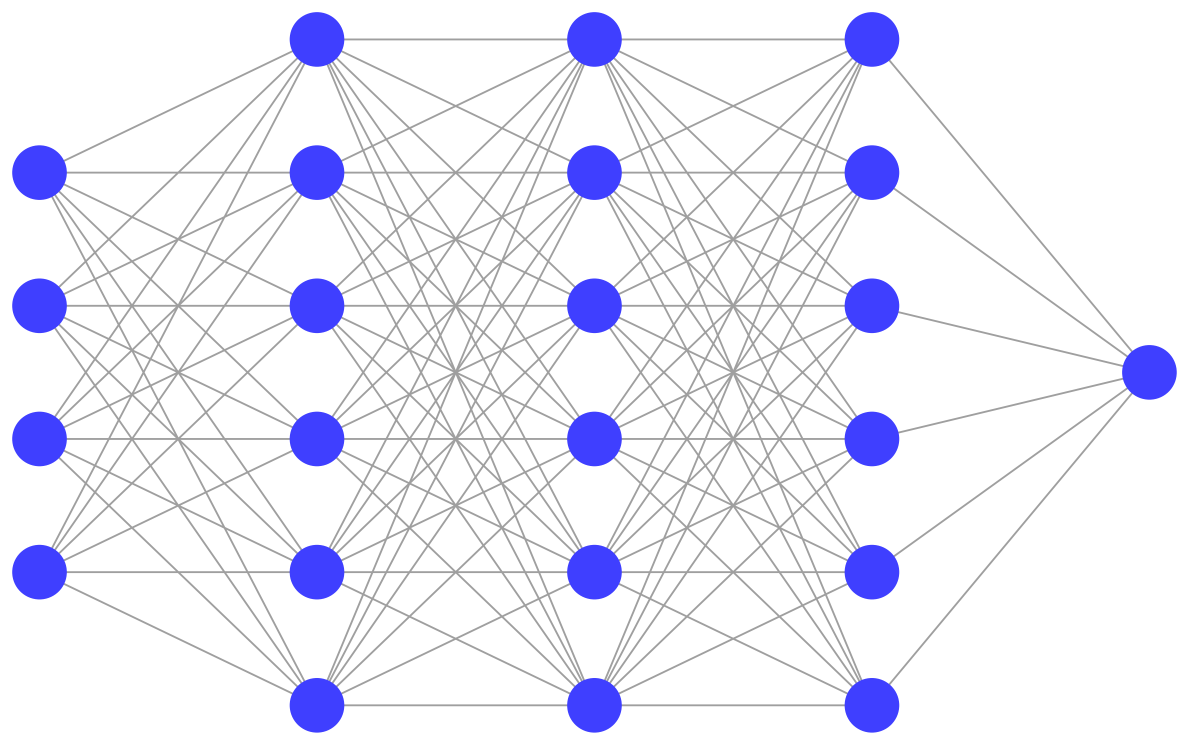
Exploiting the Gaussian equivalence



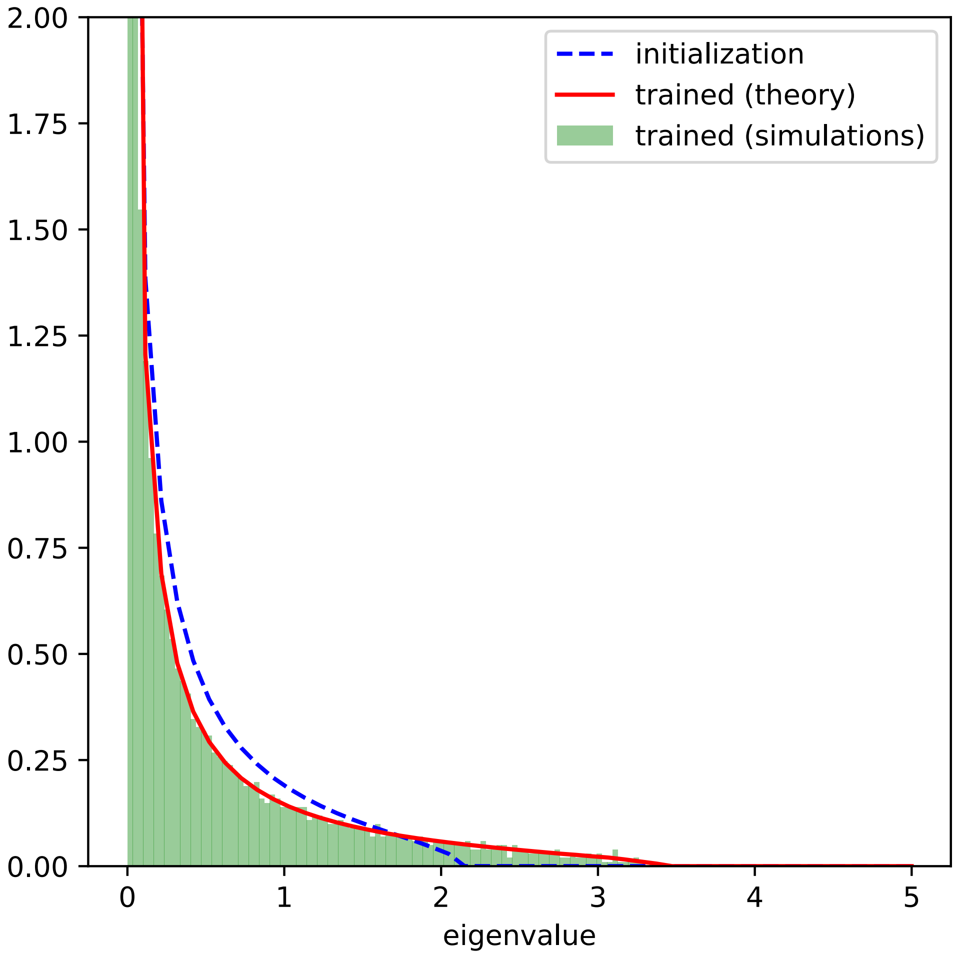
Logistic regression $\sigma(x) = \tanh(x)$

Extensions and recent developments

- Multilayer random feature models [Bosch, Panahi, Hassibi ’23]
- Data augmentation via noise injection [Dhiallah & Lu ’21]
- Beyond random feature models [Ba et al., ’22], [Cui et al., ’24], [Dandi et al, ’25]



A few gradient updates to the weight matrices:



Why does Gaussian equivalence work?

Why does Gaussian equivalence work?

For simplicity, consider ***one-hidden layer*** $\sigma(WX)$

Universality of random feature models

***Nonlinear* feature map:**

$$A = \sigma(WX)$$

***Noisy linear* feature map:**

$$B = \mu_0 1_{p\times n} + \mu_1 WX + \mu_2 Z$$

where $z_{ij} \sim_{\text{iid}} \mathcal{N}(0,1)$

Universality of random feature models

***Nonlinear* feature map:**

$$A = \sigma(WX)$$

***Noisy linear* feature map:**

$$B = \mu_0 1_{p\times n} + \mu_1 WX + \mu_2 Z$$

where $z_{ij} \sim_{\text{iid}} \mathcal{N}(0,1)$

Matching of the first two moments: for ***generic*** W

$$\mathbb{E}[\boldsymbol{a}_i] \approx \mathbb{E}[\boldsymbol{b}_i] \quad \text{and} \quad \mathbb{E}[\boldsymbol{a}_i\boldsymbol{a}_i^\top] \approx \mathbb{E}[\boldsymbol{b}_i\boldsymbol{b}_i^\top]$$

Universality of random feature models

***Nonlinear* feature map:**

$$A = \sigma(WX)$$

***Noisy linear* feature map:**

$$B = \mu_0 1_{p\times n} + \mu_1 WX + \mu_2 Z$$
where $z_{ij} \sim_{\text{iid}} \mathcal{N}(0,1)$

$$\left\| WW^\top - \boldsymbol{I} \right\|_\infty = \mathcal{O}\left(\frac{\text{polylog} d}{\sqrt{d}}\right)$$

Matching of the first two moments: for ***generic*** W

$$\mathbb{E}[\boldsymbol{a}_i] \approx \mathbb{E}[\boldsymbol{b}_i] \quad \text{and} \quad \mathbb{E}[\boldsymbol{a}_i\boldsymbol{a}_i^\top] \approx \mathbb{E}[\boldsymbol{b}_i\boldsymbol{b}_i^\top]$$

Proving the Gaussian equivalence conjecture

Assumptions:

- Convex loss functions $\ell(x; y)$ with bounded third derivatives
- Strongly convex regularizer (e.g. $\frac{\lambda}{2}\|\boldsymbol{w}\|^2$ for some $\lambda > 0$)
- Random weight matrix W with independent Gaussian entries
- The activation function $\sigma(x)$ has bounded third derivatives and it is an ***odd function***

Hu & Lu, *IEEE Trans. Inf. Theory*, arXiv:2009.07669

Proving the Gaussian equivalence conjecture

Theorem (Hu and Lu, '20)**:**

For any $\varepsilon \in (0,1)$ and constant c, we have

$$\mathbb{P}\left(\left|\mathcal{E}_{\mathrm{train}}(A)/p - c\right| \geq 2\varepsilon\right) \leq \mathbb{P}\left(\left|\mathcal{E}_{\mathrm{train}}(B)/p - c\right| \geq \varepsilon\right) + \frac{\mathrm{polylog}\, p}{\varepsilon\sqrt{p}}$$

and

$$\mathbb{P}\left(\left|\mathcal{E}_{\mathrm{train}}(B)/p - c\right| \geq 2\varepsilon\right) \leq \mathbb{P}\left(\left|\mathcal{E}_{\mathrm{train}}(A)/p - c\right| \geq \varepsilon\right) + \frac{\mathrm{polylog}\, p}{\varepsilon\sqrt{p}}$$

Consequently,

$$\mathcal{E}_{\mathrm{train}}(A)/p \xrightarrow{\mathcal{P}} c \qquad \textit{if and only if} \qquad \mathcal{E}_{\mathrm{train}}(B)/p \xrightarrow{\mathcal{P}} c$$

Similar result for the test errors.

[Hu & Lu, arXiv:2009.07669]

See also [Montanari and Saeed, 2022] for universality of free energy

Proof idea (sketch)

[Hu & Lu, arXiv:2009.07669]

Ensemble A: $\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3, \ldots, \boldsymbol{a}_{n-1}, \boldsymbol{a}_n$

Ensemble B: $\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3, \ldots, \boldsymbol{b}_{n-1}, \boldsymbol{b}_n$

See also [Panahi & Hassibi, '17], [Abbasi, Salehi, Hassibi, '19]

Proof idea (sketch)

[Hu & Lu, arXiv:2009.07669]

Ensemble A: $\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3, \ldots, \boldsymbol{a}_{n-1}, \boldsymbol{a}_n$

$$\downarrow$$

$\boldsymbol{b}_1, \boldsymbol{a}_2, \boldsymbol{a}_3, \ldots, \boldsymbol{a}_{n-1}, \boldsymbol{a}_n$

Ensemble B: $\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3, \ldots, \boldsymbol{b}_{n-1}, \boldsymbol{b}_n$

See also [Panahi & Hassibi, '17], [Abbasi, Salehi, Hassibi, '19]

Proof idea (sketch)

[Hu & Lu, arXiv:2009.07669]

Ensemble A: $\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3, \ldots, \boldsymbol{a}_{n-1}, \boldsymbol{a}_n$

$$\Downarrow$$

$\boldsymbol{b}_1, \boldsymbol{a}_2, \boldsymbol{a}_3, \ldots, \boldsymbol{a}_{n-1}, \boldsymbol{a}_n$

$$\Downarrow$$

$\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{a}_3, \ldots, \boldsymbol{a}_{n-1}, \boldsymbol{a}_n$

Ensemble B: $\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3, \ldots, \boldsymbol{b}_{n-1}, \boldsymbol{b}_n$

See also [Panahi & Hassibi, ’17], [Abbasi, Salehi, Hassibi, ’19]

Proof idea (sketch)

[Hu & Lu, arXiv:2009.07669]

Ensemble A: $\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3, \ldots, \boldsymbol{a}_{n-1}, \boldsymbol{a}_n$

$$\downarrow$$

$$\boldsymbol{b}_1, \boldsymbol{a}_2, \boldsymbol{a}_3, \ldots, \boldsymbol{a}_{n-1}, \boldsymbol{a}_n$$

$$\downarrow$$

$$\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{a}_3, \ldots, \boldsymbol{a}_{n-1}, \boldsymbol{a}_n$$

$$\downarrow$$

$$\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3, \ldots, \boldsymbol{a}_{n-1}, \boldsymbol{a}_n$$

Ensemble B: $\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3, \ldots, \boldsymbol{b}_{n-1}, \boldsymbol{b}_n$

See also [Panahi & Hassibi, ’17], [Abbasi, Salehi, Hassibi, ’19]

Proof idea (sketch)

[Hu & Lu, arXiv:2009.07669]

Ensemble A: $\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3, \ldots, \boldsymbol{a}_{n-1}, \boldsymbol{a}_n$

$$\downarrow$$

$$\boldsymbol{b}_1, \boldsymbol{a}_2, \boldsymbol{a}_3, \ldots, \boldsymbol{a}_{n-1}, \boldsymbol{a}_n$$

$$\downarrow$$

$$\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{a}_3, \ldots, \boldsymbol{a}_{n-1}, \boldsymbol{a}_n$$

$$\downarrow$$

$$\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3, \ldots, \boldsymbol{a}_{n-1}, \boldsymbol{a}_n$$

$$\downarrow$$

$$\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3, \ldots, \boldsymbol{b}_{n-1}, \boldsymbol{a}_n$$

Ensemble B: $\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3, \ldots, \boldsymbol{b}_{n-1}, \boldsymbol{b}_n$

See also [Panahi & Hassibi, '17], [Abbasi, Salehi, Hassibi, '19]

Proof idea (sketch)

[Hu & Lu, arXiv:2009.07669]

Ensemble A: $\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3, \ldots, \boldsymbol{a}_{n-1}, \boldsymbol{a}_n$

$$\downarrow$$

$$\boldsymbol{b}_1, \boldsymbol{a}_2, \boldsymbol{a}_3, \ldots, \boldsymbol{a}_{n-1}, \boldsymbol{a}_n$$

$$\downarrow$$

$$\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{a}_3, \ldots, \boldsymbol{a}_{n-1}, \boldsymbol{a}_n$$

$$\downarrow$$

$$\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3, \ldots, \boldsymbol{a}_{n-1}, \boldsymbol{a}_n$$

$$\downarrow$$

$$\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3, \ldots, \boldsymbol{b}_{n-1}, \boldsymbol{a}_n$$

$$\downarrow$$

Ensemble B: $\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3, \ldots, \boldsymbol{b}_{n-1}, \boldsymbol{b}_n$

See also [Panahi & Hassibi, '17], [Abbasi, Salehi, Hassibi, '19]

Proof idea (sketch)

[Hu & Lu, arXiv:2009.07669]

Ensemble A: $\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3, \ldots, \boldsymbol{a}_{n-1}, \boldsymbol{a}_n$

$$\downarrow$$

$\boldsymbol{b}_1, \boldsymbol{a}_2, \boldsymbol{a}_3, \ldots, \boldsymbol{a}_{n-1}, \boldsymbol{a}_n$

Number of swaps: n

$$\downarrow$$

$\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{a}_3, \ldots, \boldsymbol{a}_{n-1}, \boldsymbol{a}_n$

$$\downarrow$$

$\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3, \ldots, \boldsymbol{a}_{n-1}, \boldsymbol{a}_n$

$$\downarrow$$

$\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3, \ldots, \boldsymbol{b}_{n-1}, \boldsymbol{a}_n$

$$\downarrow$$

Ensemble B: $\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3, \ldots, \boldsymbol{b}_{n-1}, \boldsymbol{b}_n$

See also [Panahi & Hassibi, '17], [Abbasi, Salehi, Hassibi, '19]

Proof idea (sketch)

[Hu & Lu, arXiv:2009.07669]

Ensemble A: $\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3, \ldots, \boldsymbol{a}_{n-1}, \boldsymbol{a}_n$

$$\downarrow$$

$\boldsymbol{b}_1, \boldsymbol{a}_2, \boldsymbol{a}_3, \ldots, \boldsymbol{a}_{n-1}, \boldsymbol{a}_n$

$$\downarrow$$

$\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{a}_3, \ldots, \boldsymbol{a}_{n-1}, \boldsymbol{a}_n$

$$\downarrow$$

$\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3, \ldots, \boldsymbol{a}_{n-1}, \boldsymbol{a}_n$

$$\downarrow$$

$\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3, \ldots, \boldsymbol{b}_{n-1}, \boldsymbol{a}_n$

$$\downarrow$$

Ensemble B: $\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3, \ldots, \boldsymbol{b}_{n-1}, \boldsymbol{b}_n$

Number of swaps: n

Need to show each swap incurs a difference of size $o(n^{-1})$

See also [Panahi & Hassibi, ’17], [Abbasi, Salehi, Hassibi, ’19]

Proof idea (sketch)

[Hu & Lu, arXiv:2009.07669]

Ensemble A: $\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3, \ldots, \boldsymbol{a}_{n-1}, \boldsymbol{a}_n$

$$\downarrow$$

$\boldsymbol{b}_1, \boldsymbol{a}_2, \boldsymbol{a}_3, \ldots, \boldsymbol{a}_{n-1}, \boldsymbol{a}_n$

$$\downarrow$$

$\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{a}_3, \ldots, \boldsymbol{a}_{n-1}, \boldsymbol{a}_n$

$$\downarrow$$

$\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3, \ldots, \boldsymbol{a}_{n-1}, \boldsymbol{a}_n$

$$\downarrow$$

$\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3, \ldots, \boldsymbol{b}_{n-1}, \boldsymbol{a}_n$

$$\downarrow$$

Ensemble B: $\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3, \ldots, \boldsymbol{b}_{n-1}, \boldsymbol{b}_n$

Number of swaps: n

Need to show each swap incurs a difference of size $o(n^{-1})$

Main technical tools: ***Lindeberg's approach***, leave-one-out analysis, Stein's method for a central limit theorem for weakly correlated random variables

See also [Panahi & Hassibi, '17], [Abbasi, Salehi, Hassibi, '19]

Key ingredient: a central limit theorem

***Nonlinear* feature map:**

$$\boldsymbol{a} = \sigma(\boldsymbol{F}\boldsymbol{g})$$

***Noisy linear* feature map:**

$$\boldsymbol{b} = \mu_0 \mathbf{1} + \mu_1 \boldsymbol{F}\boldsymbol{g} + \mu_2 \boldsymbol{z}$$

where $z \sim \mathcal{N}(0, \boldsymbol{I}_p)$

Key ingredient: a central limit theorem

***Nonlinear* feature map:**

$$\boldsymbol{a} = \sigma(\boldsymbol{F}\boldsymbol{g})$$

***Noisy linear* feature map:**

$$\boldsymbol{b} = \mu_0 \mathbf{1} + \mu_1 \boldsymbol{F}\boldsymbol{g} + \mu_2 \boldsymbol{z}$$

where $z \sim \mathcal{N}(0, \boldsymbol{I}_p)$

A ***central limit theorem***: [Goldt et al, '20], [Hu, Lu '20]

For any fixed $\boldsymbol{w} \in \mathbb{R}^p$ with $\|\boldsymbol{w}\|_\infty \leq \mathrm{polylog}\ p$,

$$\frac{1}{\sqrt{p}} \boldsymbol{w}^\top \boldsymbol{a} \overset{\text{Law}}{\approx} \frac{1}{\sqrt{p}} \boldsymbol{w}^\top \boldsymbol{b}$$

Key ingredient: a central limit theorem

Nonlinear feature map:

$$\boldsymbol{a} = \sigma(\boldsymbol{F}\boldsymbol{g})$$

Noisy linear feature map:

$$\boldsymbol{b} = \mu_0 \mathbf{1} + \mu_1 \boldsymbol{F}\boldsymbol{g} + \mu_2 \boldsymbol{z}$$

where $z \sim \mathcal{N}(0, \boldsymbol{I}_p)$

A ***central limit theorem***: [Goldt et al, '20], [Hu, Lu '20]

For any fixed $\boldsymbol{w} \in \mathbb{R}^p$ with $\|\boldsymbol{w}\|_\infty \leq \mathrm{polylog}\ p$,

$$\frac{1}{\sqrt{p}} \boldsymbol{w}^\top \boldsymbol{a} \overset{\text{Law}}{\approx} \frac{1}{\sqrt{p}} \boldsymbol{w}^\top \boldsymbol{b}$$

In this lecture: a short proof based on Wiener chaos expansion