Random Matrix Theory for Machine Learning

The Mystery of Generalization: Why Does Deep Learning Work?

Fabian Pedregosa, Courtney Paquette, Tom Trogdon, Jeffrey Pennington

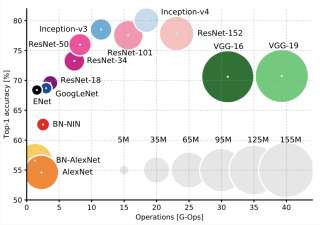
https://random-matrix-learning.github.io

Why does deep learning work?

Deep neural networks define a flexible and expressive class of functions.

State-of-the-art models have millions or billions of parameters

- Meena: 2.6 billion
- Turing NLG: 17 billion
- GPT-3: 175 billion



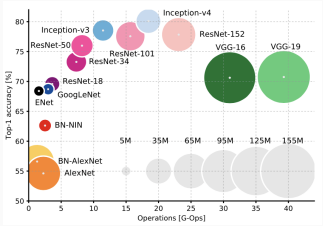
Source: [Canziani et al., 2016]

Why does deep learning work?

Deep neural networks define a flexible and expressive class of functions.

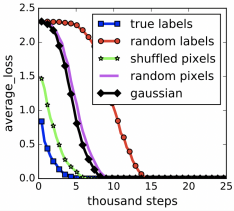
State-of-the-art models have millions or billions of parameters

- Meena: 2.6 billion
- Turing NLG: 17 billion
- GPT-3: 175 billion



Source: [Canziani et al., 2016]

Models that perform well on real data can easily memorize noise

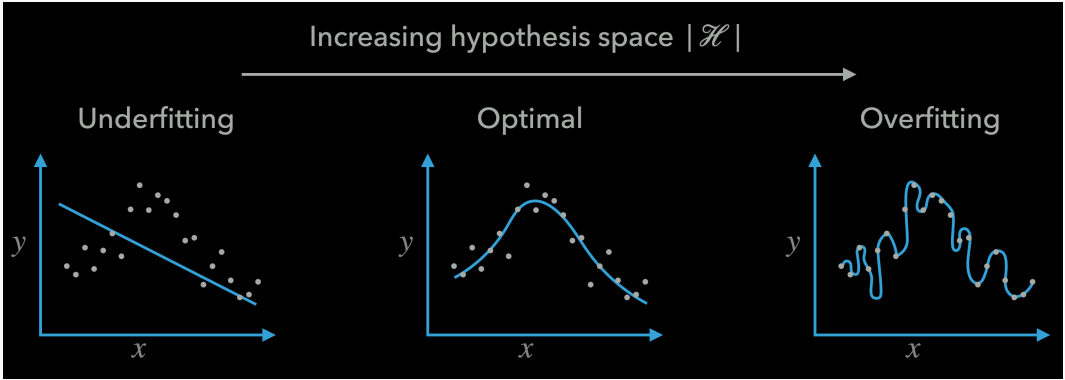


Source: [Zhang et al., 2021]

Why does deep learning work?

Deep neural networks define a flexible and expressive class of functions.

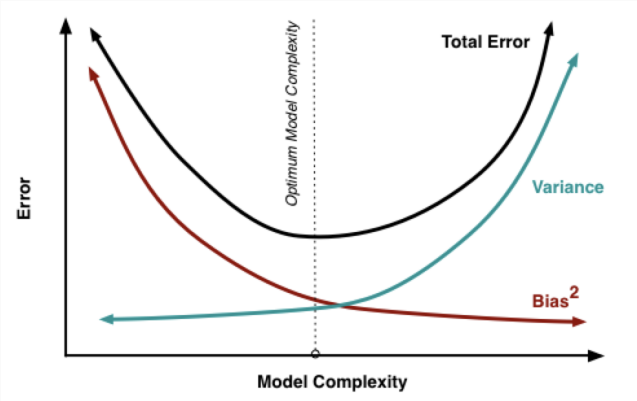
$\Rightarrow$ Standard wisdom suggests they should overfit



Why does deep learning work?

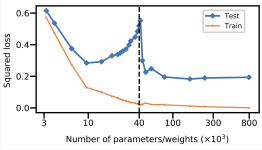
Deep neural networks define a flexible and expressive class of functions.

$\Rightarrow$ Standard wisdom suggests they should overfit

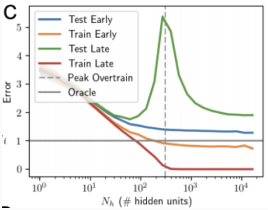


Double descent

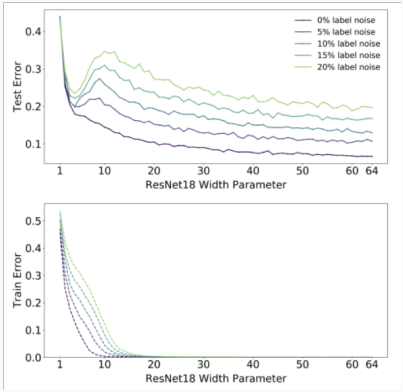
However, large neural networks do not obey the classical theory:



Source: [Belkin et al., 2019]



Source: [Advani et al., 2020]



Source: [Nakkiran et al., 2019]

The emerging paradigm of *double descent* seeks to explain this phenomenon.

Models of Double Descent

History of double descent: Kernel interpolation

1) Interpolating kernels (trained to zero error) generalize well [Belkin et al., 2018]

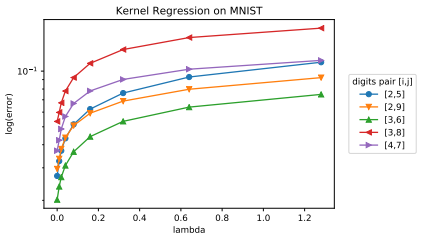
$\Rightarrow$ Double descent is not unique to deep neural networks

History of double descent: Kernel interpolation

1) Interpolating kernels (trained to zero error) generalize well [Belkin et al., 2018]

⇒ Double descent is not unique to deep neural networks

2) Kernels can implicitly regularize in high dimensions [Liang and Rakhlin, 2020]



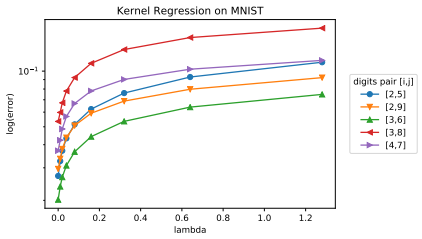
Source: [Liang and Rakhlin, 2020]

History of double descent: Kernel interpolation

1) Interpolating kernels (trained to zero error) generalize well [Belkin et al., 2018]

⇒ Double descent is not unique to deep neural networks

2) Kernels can implicitly regularize in high dimensions [Liang and Rakhlin, 2020]



Source: [Liang and Rakhlin, 2020]

3) Consistency is a high-dimensional phenomenon [Rakhlin and Zhai, 2019]:

The estimation error of the minimum-norm kernel interpolant

$$\arg\min_{f \in \mathcal{H}} \|f\|_{\mathcal{H}} \quad \text{s.t.} \quad f(x_i) = y_i\,, \quad i = 1 \dots n$$

does not converge to zero as n grows, unless d is proportional to n.

Models of double descent: High-dimensional linear regression

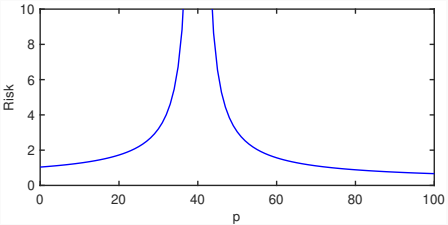
What is the simplest theoretically tractable model that exhibits double descent?

Models of double descent: High-dimensional linear regression

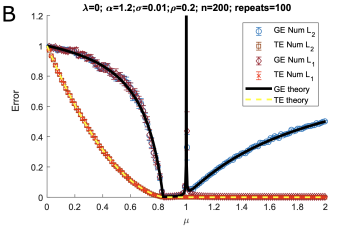
What is the simplest theoretically tractable model that exhibits double descent?

Linear regression suffices, but requires a mechanism to vary the effective number of parameters or samples:

- The size of randomly selected subsets of features [Belkin et al., 2020]
- The dimensionality of the low-variance subspace [Bartlett et al., 2020]
- The sparsity of the generative model [Mitra, 2019]



Source: [Belkin et al., 2020]



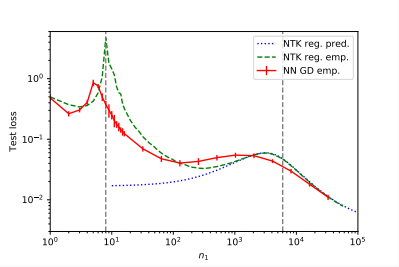
Source: [Mitra, 2019]

Models of double descent: Random feature models

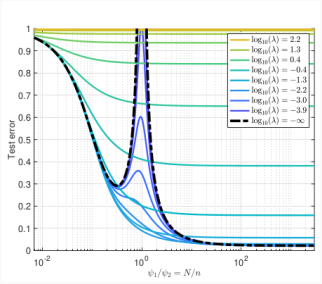
In random feature regression, the number of random features controls the model capacity, and can be tuned independently from the data.

Exact asymptotic results in high dimensions exist in many settings, including:

- Unstructured random features [Mei and Montanari, 2019]
- NTK-structured random features [Adlam and Pennington, 2020]
- Random Fourier features [Liao et al., 2020]



Source: [Adlam and Pennington, 2020]



Source: [Mei and Montanari, 2019]

Random feature models

Random feature regression: definition

Random feature regression is just linear regression on a transformed feature matrix, $F = f(\frac{1}{\sqrt{d}} WX) \in \mathbb{R}^{m \times n}$, where $W \in \mathbb{R}^{m \times d}$, $W_{ij} \sim \mathcal{N}(0, 1)$.

- Model given by $\beta^\top F$ (instead of $\beta^\top X$) — variable capacity (m vs d parameters)
- $f(\cdot)$ is a nonlinear activation function, acting elementwise
- F is equivalent to first post-activation layer of a NN at init

Random feature regression: definition

Random feature regression is just linear regression on a transformed feature matrix, $F = f(\frac{1}{\sqrt{d}} WX) \in \mathbb{R}^{m \times n}$, where $W \in \mathbb{R}^{m \times d}$, $W_{ij} \sim \mathcal{N}(0, 1)$.

- Model given by $\beta^\top F$ (instead of $\beta^\top X$) — variable capacity (m vs d parameters)
- $f(\cdot)$ is a nonlinear activation function, acting elementwise
- F is equivalent to first post-activation layer of a NN at init

For targets $Y \in \mathbb{R}^{1 \times n}$, the ridge-regularized loss is,

$$L(\beta; X) = \|Y - \tfrac{1}{\sqrt{m}} \beta^\top F\|_2^2 + \lambda \|\beta\|_2^2 \,,$$

and the optimal regression coefficients $\hat{\beta}$ are given by,

$$\hat{\beta} = \tfrac{1}{\sqrt{m}} Y (K + \lambda I_n)^{-1} F^\top \,, \quad K = \tfrac{1}{m} F^\top F \,.$$

Random feature regression: definition

Random feature regression is just linear regression on a transformed feature matrix, $F = f(\frac{1}{\sqrt{d}} WX) \in \mathbb{R}^{m \times n}$, where $W \in \mathbb{R}^{m \times d}$, $W_{ij} \sim \mathcal{N}(0, 1)$.

- Model given by $\beta^\top F$ (instead of $\beta^\top X$) — variable capacity (m vs d parameters)
- $f(\cdot)$ is a nonlinear activation function, acting elementwise
- F is equivalent to first post-activation layer of a NN at init

For targets $Y \in \mathbb{R}^{1 \times n}$, the ridge-regularized loss is,

$$L(\beta; X) = \|Y - \tfrac{1}{\sqrt{m}} \beta^\top F\|_2^2 + \lambda \|\beta\|_2^2 \,,$$

and the optimal regression coefficients $\hat{\beta}$ are given by,

$$\hat{\beta} = \tfrac{1}{\sqrt{m}} Y (K + \lambda I_n)^{-1} F^\top \,, \quad K = \tfrac{1}{m} F^\top F \,.$$

Note that $Q \equiv (K + \lambda I_n)^{-1}$ is the *resolvent* of the kernel matrix K.

Random feature regression: training error

Training error is determined by the resolvent matrix $Q \equiv (K + \lambda I_n)^{-1}$:

$$\begin{aligned} E_{\text{train}}(\lambda) &= \tfrac{1}{n} \| Y - \tfrac{1}{\sqrt{m}} \hat{\boldsymbol{\beta}}^\top F \|_2^2 \\ &= \tfrac{1}{n} \| Y - \tfrac{1}{m} Y Q F^\top F \|_2^2 \\ &= \tfrac{1}{n} \| Y - Y Q K \|_2^2 \\ &= \tfrac{1}{n} \| Y (I_n - Q K) \|_2^2 \\ &= \lambda^2 \tfrac{1}{n} \| Y Q \|_2^2 \,, \end{aligned}$$

where we used that $I_n - QK = I_n - Q(Q^{-1} - \lambda I_n) = I_n - (I_n - \lambda Q) = \lambda Q$.

Random feature regression: training error

Training error is determined by the resolvent matrix $Q \equiv (K + \lambda I_n)^{-1}$:

$$\begin{aligned} E_{\text{train}}(\lambda) &= \tfrac{1}{n} \| Y - \tfrac{1}{\sqrt{m}} \hat{\beta}^\top F \|_2^2 \\ &= \tfrac{1}{n} \| Y - \tfrac{1}{m} Y Q F^\top F \|_2^2 \\ &= \tfrac{1}{n} \| Y - Y Q K \|_2^2 \\ &= \tfrac{1}{n} \| Y (I_n - Q K) \|_2^2 \\ &= \lambda^2 \tfrac{1}{n} \| Y Q \|_2^2 \,, \end{aligned}$$

where we used that $I_n - QK = I_n - Q(Q^{-1} - \lambda I_n) = I_n - (I_n - \lambda Q) = \lambda Q$.

So we see that the training error measures the alignment between the resolvent and the label vector.

What about the test error?

Aside: Generalized cross validation (GCV)

A model's performance on the training set, or subsets thereof, can be useful for estimating its performance on the test set.

- Leave-one-out cross validation (LOOCV)

$$E_{LOOCV}(\lambda) = \frac{1}{n}\|\boldsymbol{YQ} \cdot \mathrm{diag}(\boldsymbol{Q})^{-1}\|_2^2$$

- Generalized cross validation (GCV)

$$E_{GCV}(\lambda) = \frac{1}{n}\|\boldsymbol{YQ}\|_2^2 / (\frac{1}{n}\mathrm{tr}(\boldsymbol{Q}))^2$$

Aside: Generalized cross validation (GCV)

A model's performance on the training set, or subsets thereof, can be useful for estimating its performance on the test set.

- Leave-one-out cross validation (LOOCV)

$$E_{LOOCV}(\lambda) = \frac{1}{n}\|\boldsymbol{YQ} \cdot \mathrm{diag}(\boldsymbol{Q})^{-1}\|_2^2$$

- Generalized cross validation (GCV)

$$E_{GCV}(\lambda) = \frac{1}{n}\|\boldsymbol{YQ}\|_2^2 / (\frac{1}{n}\mathrm{tr}(\boldsymbol{Q}))^2$$

In certain high-dimensional limits, $E_{GCV}(\lambda) = E_{LOOCV}(\lambda) = E_{\text{test}}(\lambda)$:

- Ridge regression [Hastie et al., 2019]
- Kernel ridge regression [Jacot et al., 2020]
- Random feature regression [Adlam and Pennington, 2020]

Random feature regression: high-dimensional asymptotics

To develop an analytical model of double descent, we study the high-dimensional asymptotics:

$$m, d, n \to \infty \quad \text{such that} \quad \phi \equiv \frac{d}{n}, \psi \equiv \frac{d}{m} \quad \text{are constant.}$$

Random feature regression: high-dimensional asymptotics

To develop an analytical model of double descent, we study the high-dimensional asymptotics:

$$m, d, n \to \infty \quad \text{such that} \quad \phi \equiv \frac{d}{n}, \psi \equiv \frac{d}{m} \quad \text{are constant.}$$

In this limit, only *linear* functions of the data can be learned.

- Intuition: only enough constraints to disambiguate linear combinations of features.
- Nonlinear target function behaves like linear function plus noise

Random feature regression: high-dimensional asymptotics

To develop an analytical model of double descent, we study the high-dimensional asymptotics:

$$m, d, n \to \infty \quad \text{such that} \quad \phi \equiv \frac{d}{n}\,, \psi \equiv \frac{d}{m} \quad \text{are constant.}$$

In this limit, only *linear* functions of the data can be learned.

- Intuition: only enough constraints to disambiguate linear combinations of features.
- Nonlinear target function behaves like linear function plus noise

Therefore it suffices to consider labels given by

$$Y = \frac{1}{\sqrt{d}} \boldsymbol{\beta}^{*\top} \boldsymbol{X} + \boldsymbol{\varepsilon}\,, \quad \varepsilon_i \sim \mathcal{N}(0, \sigma_\varepsilon^2)\,.$$

Random feature regression: high-dimensional asymptotics

To develop an analytical model of double descent, we study the high-dimensional asymptotics:

$$m, d, n \to \infty \quad \text{such that} \quad \phi \equiv \tfrac{d}{n}\,, \psi \equiv \tfrac{d}{m} \quad \text{are constant.}$$

In this limit, only *linear* functions of the data can be learned.

- Intuition: only enough constraints to disambiguate linear combinations of features.
- Nonlinear target function behaves like linear function plus noise

Therefore it suffices to consider labels given by

$$Y = \tfrac{1}{\sqrt{d}} \boldsymbol{\beta}^{*\top} X + \boldsymbol{\varepsilon}\,, \quad \varepsilon_i \sim \mathcal{N}(0, \sigma_\varepsilon^2)\,.$$

For simplicity, we focus on the specific setting in which,

$$X_{ij} \sim \mathcal{N}(0,1) \quad \text{and} \quad \boldsymbol{\beta}^* \sim \mathcal{N}(0, I_d)\,.$$

Random feature regression: test error

In the high-dimensional asymptotic setup from above, the random feature test error can be written as,

$$\begin{aligned} E_{\text{test}}(\lambda) &= E_{\text{GCV}}(\lambda) \\ &= \lim_{n\to\infty} \tfrac{1}{n}\|YQ\|_2^2/(\tfrac{1}{n}\text{tr}(Q))^2 \\ &= \lim_{n\to\infty} \tfrac{1}{n}\|(\tfrac{1}{\sqrt{d}}\beta^{*\top}X + \varepsilon)Q\|_2^2/(\tfrac{1}{n}\text{tr}(Q))^2 \\ &= \lim_{n\to\infty} \tfrac{1}{n}\text{tr}[(\sigma_\varepsilon^2 I_n + \tfrac{1}{d}X^\top X)Q^2]/(\tfrac{1}{n}\text{tr}(Q))^2 \\ &\equiv -\frac{\sigma_\varepsilon^2\tau_1'(\lambda) + \tau_2'(\lambda)}{\tau_1(\lambda)^2}\,, \end{aligned}$$

where we used that $\frac{\partial}{\partial\lambda}Q = -Q^2$, and we defined

$$\tau_1 = \lim_{n\to\infty} \tfrac{1}{n}\text{tr}(Q) \quad \text{and} \quad \tau_2 = \lim_{n\to\infty} \tfrac{1}{n}\text{tr}(\tfrac{1}{d}X^\top XQ)\,.$$

Computing the asymptotic test error

To compute the test error, we need:

$$\tau_1 = \lim_{n\to\infty} \tfrac{1}{n}\mathrm{tr}(K + \lambda I_n)^{-1} \quad \text{and} \quad \tau_2 = \lim_{n\to\infty} \tfrac{1}{n}\mathrm{tr}(\tfrac{1}{d}X^\top X(K + \lambda I_n)^{-1}) \,.$$

Computing the asymptotic test error

To compute the test error, we need:

$$\tau_1 = \lim_{n\to\infty} \tfrac{1}{n}\mathrm{tr}(K + \lambda I_n)^{-1} \quad \text{and} \quad \tau_2 = \lim_{n\to\infty} \tfrac{1}{n}\mathrm{tr}(\tfrac{1}{d}X^\top X(K + \lambda I_n)^{-1}) \,.$$

Recalling the definition of K,

$$K = \tfrac{1}{m}F^\top F \,, \quad F = f(\tfrac{1}{\sqrt{d}}WX) \,,$$

it is evident that the entries of F are nonlinearly dependent.

- Cannot simply utilize standard results for Wishart matrices
- Stieltjes transform is insufficient for τ_2

Computing the asymptotic test error

To compute the test error, we need:

$$\tau_1 = \lim_{n\to\infty} \tfrac{1}{n}\mathrm{tr}(K + \lambda I_n)^{-1} \quad \text{and} \quad \tau_2 = \lim_{n\to\infty} \tfrac{1}{n}\mathrm{tr}(\tfrac{1}{d}X^\top X(K + \lambda I_n)^{-1}) \,.$$

Recalling the definition of K,

$$K = \tfrac{1}{m}F^\top F \,, \quad F = f(\tfrac{1}{\sqrt{d}}WX) \,,$$

it is evident that the entries of F are nonlinearly dependent.

- Cannot simply utilize standard results for Wishart matrices
- Stieltjes transform is insufficient for τ_2

These technical challenges can be overcome with two tricks:

1. Constructing an equivalent Gaussian linearized model
2. Analyzing a suitably augmented resolvent

Computing the asymptotic test error: Gaussian equivalents

The nonlinear dependencies in $F = f(\frac{1}{\sqrt{d}}WX)$ complicate the analysis.

Can we identify a simpler matrix in the same universality class?

Computing the asymptotic test error: Gaussian equivalents

The nonlinear dependencies in $F = f(\frac{1}{\sqrt{d}} WX)$ complicate the analysis.

Can we identify a simpler matrix in the same universality class?

There exist constants c_1 and c_2 such that

$$F \cong F_{\text{lin}} \equiv c_1 \tfrac{1}{\sqrt{d}} WX + c_2 \Theta\,, \quad \Theta_{ij} \sim \mathcal{N}(0,1)\,,$$

where $F \cong F_{\text{lin}}$ indicates the two matrices share all statistics relevant for computing the test error:

$$\tau_1 = \lim_{n\to\infty} \tfrac{1}{n} \text{tr}(\tfrac{1}{m} F^\top F + \lambda I_n)^{-1} = \lim_{n\to\infty} \tfrac{1}{n} \text{tr}(\tfrac{1}{m} F_{\text{lin}}^\top F_{\text{lin}} + \lambda I_n)^{-1}$$

$$\tau_2 = \lim_{n\to\infty} \tfrac{1}{n} \text{tr}(\tfrac{1}{d} X^\top X (\tfrac{1}{m} F^\top F + \lambda I_n)^{-1}) = \lim_{n\to\infty} \tfrac{1}{n} \text{tr}(\tfrac{1}{d} X^\top X (\tfrac{1}{m} F_{\text{lin}}^\top F_{\text{lin}} + \lambda I_n)^{-1})$$

Computing the asymptotic test error: Gaussian equivalents

The nonlinear dependencies in $F = f(\frac{1}{\sqrt{d}}WX)$ complicate the analysis.

Can we identify a simpler matrix in the same universality class?

There exist constants c_1 and c_2 such that

$$F \cong F_{\text{lin}} \equiv c_1 \frac{1}{\sqrt{d}} WX + c_2 \Theta\,, \quad \Theta_{ij} \sim \mathcal{N}(0,1)\,,$$

where $F \cong F_{\text{lin}}$ indicates the two matrices share all statistics relevant for computing the test error:

$$\tau_1 = \lim_{n\to\infty} \tfrac{1}{n}\text{tr}(\tfrac{1}{m}F^\top F + \lambda I_n)^{-1} = \lim_{n\to\infty} \tfrac{1}{n}\text{tr}(\tfrac{1}{m}F_{\text{lin}}^\top F_{\text{lin}} + \lambda I_n)^{-1}$$
$$\tau_2 = \lim_{n\to\infty} \tfrac{1}{n}\text{tr}(\tfrac{1}{d}X^\top X(\tfrac{1}{m}F^\top F + \lambda I_n)^{-1}) = \lim_{n\to\infty} \tfrac{1}{n}\text{tr}(\tfrac{1}{d}X^\top X(\tfrac{1}{m}F_{\text{lin}}^\top F_{\text{lin}} + \lambda I_n)^{-1})$$

How can we compute these traces? Need to augment the resolvent.

Computing the asymptotic test error: resolvent method

Recall from Part 2 that the resolvent method identifies consistency relations between suitably chosen submatrices of the resolvent.

Here we can undertake a similar analysis as in Part 2, but now on an augmented matrix,

$$H = \begin{bmatrix} \lambda I_n & \frac{1}{\sqrt{m}} F_{\text{lin}}^\top \\ \frac{1}{\sqrt{m}} F_{\text{lin}} & -I_m \end{bmatrix},$$

which encodes the resolvent through $Q = (K + \lambda I_n)^{-1} = [H^{-1}]_{1:n,1:n}$.

Computing the asymptotic test error: resolvent method

Recall from Part 2 that the resolvent method identifies consistency relations between suitably chosen submatrices of the resolvent.

Here we can undertake a similar analysis as in Part 2, but now on an augmented matrix,

$$H = \begin{bmatrix} \lambda I_n & \frac{1}{\sqrt{m}} F_{\text{lin}}^\top \\ \frac{1}{\sqrt{m}} F_{\text{lin}} & -I_m \end{bmatrix},$$

which encodes the resolvent through $Q = (K + \lambda I_n)^{-1} = [H^{-1}]_{1:n,1:n}$.

To derive consistency relations, we consider two submatrices: $H^{(1)}$ (leaving out row/column 1), and $H^{(n+1)}$ (leaving out row/column $n+1$).

As before, we use the Sherman-Morrison formula to compute $[H^{(1)}]^{-1}$ and $[H^{(n+1)}]^{-1}$, and relate them to Q and rows/columns of F_{lin}.

Straightforward concentration arguments eventually lead to coupled self-consistent equations for τ_1 and τ_2 [Adlam et al., 2019].

Computing the asymptotic test error: free probability

An alternative augmentation of the resolvent completely linearizes the dependence on the random matrices:

$$M = \begin{bmatrix} \lambda I_n & \frac{c_2}{m}\Theta^\top & \frac{c_1}{\sqrt{dm}}X^\top & 0 \\ c_2\Theta & -I_m & 0 & \frac{c_1}{\sqrt{d}}W \\ 0 & W^\top & -I_d & 0 \\ X & 0 & 0 & -I_d \end{bmatrix},$$

where the Schur complement formula now gives,

$$\tau_1 = \lim_{n\to\infty} \tfrac{1}{n}\mathrm{tr}([M^{-1}]_{1,1}), \quad \text{and} \quad \tau_2 = \lim_{n\to\infty} \tfrac{1}{n}\mathrm{tr}([M^{-1}]_{4,3}).$$

The asymptotic blockwise traces $\mathrm{tr}([M^{-1}]_{a,b})$ can themselves be computed using free probability [Adlam and Pennington, 2020].

Computing the asymptotic test error: free probability

M is linear in the random matrices X, W, and Θ:

$$M = \begin{bmatrix} \lambda I_n & 0 & 0 & 0 \\ 0 & -I_m & 0 & 0 \\ 0 & 0 & -I_d & 0 \\ 0 & 0 & 0 & -I_d \end{bmatrix} + \begin{bmatrix} 0 & 0 & \frac{c_1}{\sqrt{dm}} X^\top & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ X & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{c_1}{\sqrt{d}} W \\ 0 & W^\top & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & \frac{c_2}{m} \Theta^\top & 0 & 0 \\ c_2 \Theta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Can we compute the blockwise traces with free probability via the R-transform?

Computing the asymptotic test error: free probability

M is linear in the random matrices X, W, and Θ:

$$M = \begin{bmatrix} \lambda I_n & 0 & 0 & 0 \\ 0 & -I_m & 0 & 0 \\ 0 & 0 & -I_d & 0 \\ 0 & 0 & 0 & -I_d \end{bmatrix} + \begin{bmatrix} 0 & 0 & \frac{c_1}{\sqrt{dm}} X^\top & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ X & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{c_1}{\sqrt{d}} W \\ 0 & W^\top & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & \frac{c_2}{m} \Theta^\top & 0 & 0 \\ c_2 \Theta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Can we compute the blockwise traces with free probability via the R-transform?

Not naively: the additive terms are independent, but not free over $\mathbb{C}$.

However, they are free over $M_4(\mathbb{C})$, and there exists a suitable *operator-valued* generalization of the R-transform that enables the necessary computations [Mingo and Speicher, 2017].

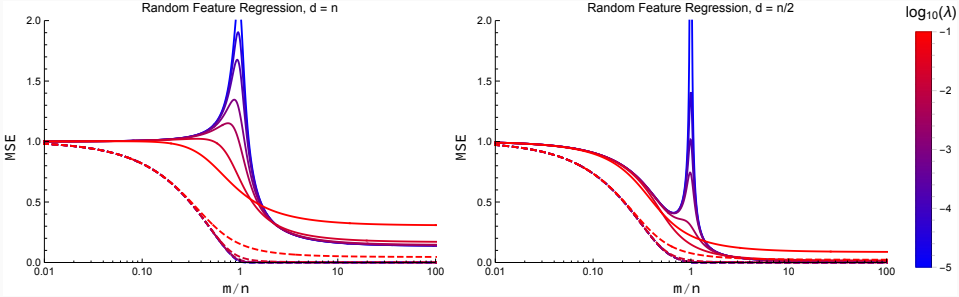
Asymptotic test error

Theorem

Let $\eta = \mathbb{E}[f(g)^2]$ and $\zeta = (\mathbb{E}[gf(g)])^2$ for $g \sim \mathcal{N}(0,1)$. Then, the asymptotic traces $\tau_1(\lambda)$ and $\tau_2(\lambda)$ are given by solutions to the polynomial system,

$$\zeta\tau_1\tau_2\left(1-\lambda\tau_1\right) = \phi/\psi\left(\zeta\tau_1\tau_2 + \phi(\tau_2-\tau_1)\right) = \left(\tau_1-\tau_2\right)\phi\left((\eta-\zeta)\tau_1 + \zeta\tau_2\right),$$

and, $E_{\text{train}} = -\lambda^2(\sigma_\varepsilon^2\tau_1' + \tau_2')$ and $E_{\text{test}} = -(\sigma_\varepsilon^2\tau_1' + \tau_2')/\tau_1^2$.



References

Ben Adlam and Jeffrey Pennington. The neural tangent kernel in high dimensions: Triple descent and a multi-scale theory of generalization. In *International Conference on Machine Learning*. PMLR, 2020. URL *https://arxiv.org/pdf/2008.06786.pdf*.

Ben Adlam, Jake Levinson, and Jeffrey Pennington. A random matrix perspective on mixtures of nonlinearities for deep learning. *arXiv preprint arXiv:1912.00827*, 2019.

Madhu S Advani, Andrew M Saxe, and Haim Sompolinsky. High-dimensional dynamics of generalization error in neural networks. *Neural Networks*, 132:428–446, 2020.

Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020.

Mikhail Belkin, Siyuan Ma, and Soumik Mandal. To understand deep learning we need to understand kernel learning. In *International Conference on Machine Learning*, pages 541–549. PMLR, 2018.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

References ii

Mikhail Belkin, Daniel Hsu, and Ji Xu. Two models of double descent for weak features. *SIAM Journal on Mathematics of Data Science*, 2(4):1167–1180, 2020.

Alfredo Canziani, Adam Paszke, and Eugenio Culurciello. An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*, 2016.

Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *arXiv preprint arXiv:1903.08560*, 2019. URL https://arxiv.org/pdf/1903.08560.pdf.

Arthur Jacot, Berfin Şimşek, Francesco Spadaro, Clément Hongler, and Franck Gabriel. Kernel alignment risk estimator: risk prediction from training data. *arXiv preprint arXiv:2006.09796*, 2020.

Tengyuan Liang and Alexander Rakhlin. Just interpolate: Kernel "ridgeless" regression can generalize. *The Annals of Statistics*, 48 (3):1329–1347, 2020.

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. *arXiv preprint arXiv:2006.05013*, 2020. URL https://arxiv.org/pdf/2006.05013.pdf.

Song Mei and Andrea Montanari. The generalization error of random features regression: Precise asymptotics and double descent curve. *arXiv preprint arXiv:1908.05355*, 2019. URL https://arxiv.org/pdf/1908.05355.pdf.

James A Mingo and Roland Speicher. *Free probability and random matrices*, volume 35. Springer, 2017.

Partha P Mitra. Understanding overfitting peaks in generalization error: Analytical risk curves for l_2 and l_1 penalized interpolation. *arXiv preprint arXiv:1906.03667*, 2019.

References iii

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *arXiv preprint arXiv:1912.02292*, 2019.

Alexander Rakhlin and Xiyu Zhai. Consistency of interpolation with laplace kernels is a high-dimensional phenomenon. In *Conference on Learning Theory*, pages 2595–2623. PMLR, 2019.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.