Lausanne Event on ML and NN Theory 2024

A Theory of Non-Linear Feature Learning with One Gradient Step in Two-Layer Neural Networks[1]

Edgar Dobriban

University of Pennsylvania

May 31, 2024

[1] ICML 2024. Slide credit: Behrad Moniri

Collaborators

Behrad Moniri

Donghwan Lee

Hamed Hassani

Introduction

Deep Learning is *very* successful.

Introduction

- It is a huge *engineering* success.

Introduction

- It is a huge *engineering* success.
- Why is it so successful? Despite fitting so many parameters and aiming to solve hugely non-convex problems?

Introduction

- It is a huge *engineering* success.
- Why is it so successful? Despite fitting so many parameters and aiming to solve hugely non-convex problems?
- Vast range of theoretical explanations have been proposed... But still no definitive answer.

Some proposed explanations

- Compositional architecture: approximate anything, $A \to B$, $B \to C \implies A \to C$

Some proposed explanations

- Compositional architecture: approximate anything, $A \to B$, $B \to C \implies A \to C$
- Algorithmic/implicit bias ensures generalization

Some proposed explanations

- Compositional architecture: approximate anything, $A \to B, B \to C \implies A \to C$

- Algorithmic/implicit bias ensures generalization

- Adapts to structure of data (low-dimensional manifold)

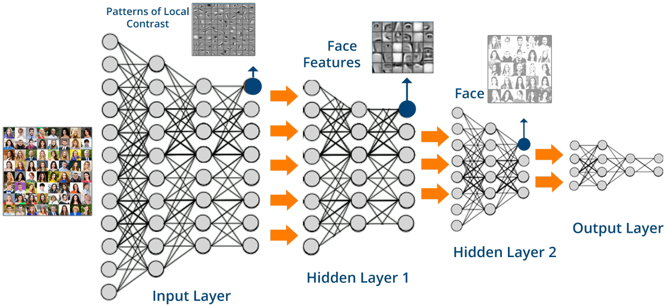
Some proposed explanations

- Compositional architecture: approximate anything, $A \to B, B \to C \implies A \to C$
- Algorithmic/implicit bias ensures generalization
- Adapts to structure of data (low-dimensional manifold)
- Feature learning

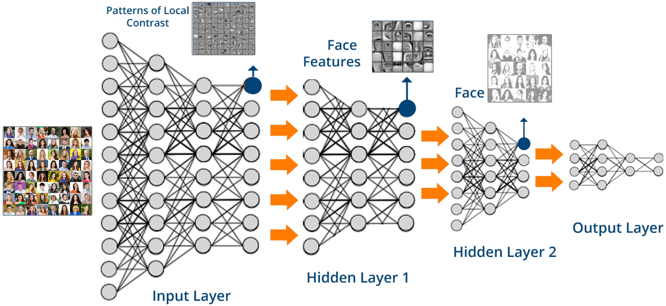
Some proposed explanations

- Compositional architecture: approximate anything, $A \to B, B \to C \implies A \to C$
- Algorithmic/implicit bias ensures generalization
- Adapts to structure of data (low-dimensional manifold)
- Feature learning
- ...

Feature Learning



Feature Learning



People also ask

Why is deep learning more effective?

Unlike traditional machine learning techniques, deep learning algorithms can automatically extract intricate patterns and features from raw data, eliminating the need for manual feature engineering. This not only saves valuable time but also enhances the efficiency and accuracy of predictive models. Feb 29, 2024

Our work

- We provide theoretical results showing *non-linear feature learning*

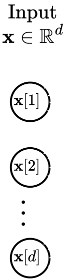
Our work

- We provide theoretical results showing *non-linear feature learning*
- Consider two-layer fully connected networks, proportional asymptotics

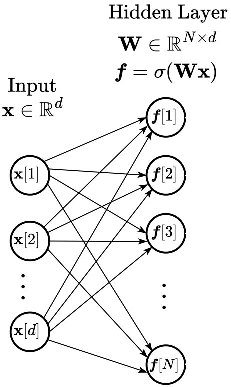
Our work

- We provide theoretical results showing *non-linear feature learning*
- Consider two-layer fully connected networks, proportional asymptotics
  1. Non-standard training, isotropic data

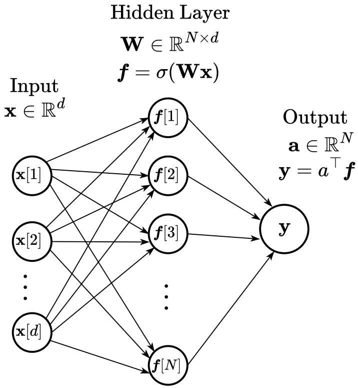
Two-Layer Neural Networks



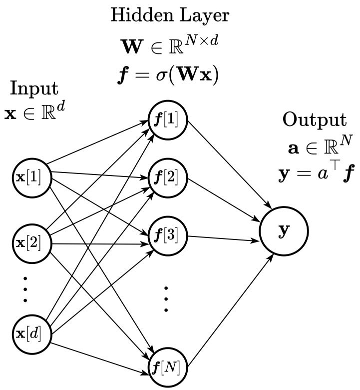
Two-Layer Neural Networks



Two-Layer Neural Networks

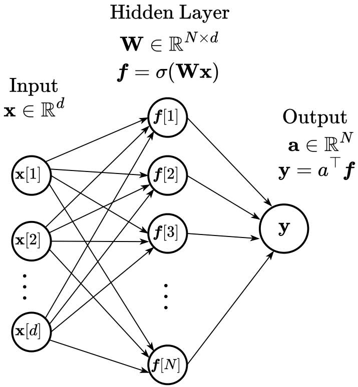


Two-Layer Neural Networks



Proportional Asymptotic Regime: $n, d, N \to \infty$ with $d/n \to \phi$ and $d/N \to \psi$.

Training and Optimization



- **Simplest Version:**

  Random Feature Model.

  (Rahimi and Recht, 2007)

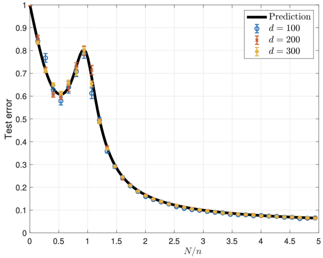
Random Feature Models

- Analysis of random feature models is popular:

Random Feature Models

- Analysis of random feature models is popular:
  - Used to study various aspects of deep learning such as double descent, robustness to adversarial attacks, privacy, fairness, OOD performance, calibration, etc.

See e.g., Mei and Montanari (2022); Gerace et al. (2020); Lin and Dobriban (2021); Lee et al. (2023); Hassani and Javanmard (2022); Bombari and Mondelli (2023); Bombari et al. (2023); Clarté et al. (2023), etc.



Double descent in random feature models (Mei and Montanari, 2022).

Random Feature Models

- Random feature models can only learn linear functions under proportional asymptotics.

  Ghorbani et al. (2021); Mei and Montanari (2022); Hu and Lu (2023),...

Random Feature Models

- Random feature models can only learn linear functions under proportional asymptotics.

  Ghorbani et al. (2021); Mei and Montanari (2022); Hu and Lu (2023),...

- **Gaussian Equivalence** El Karoui (2010); Cheng and Singer (2013); Fan and Montanari (2019); Goldt et al. (2020, 2022); Gerace et al. (2020); Mei and Montanari (2022); Pennington and Worah (2017); Adlam and Pennington (2020),...:

Random Feature Models

- Random feature models can only learn linear functions under proportional asymptotics.

  Ghorbani et al. (2021); Mei and Montanari (2022); Hu and Lu (2023),...

- **Gaussian Equivalence** El Karoui (2010); Cheng and Singer (2013); Fan and Montanari (2019); Goldt et al. (2020, 2022); Gerace et al. (2020); Mei and Montanari (2022); Pennington and Worah (2017); Adlam and Pennington (2020),...:

$$\sigma(\mathbf{W}x) = c_1\mathbf{W}x + c_2 H_2(\mathbf{W}x) + \cdots$$

$$\approx c_1\mathbf{W}x + z$$

Random Feature Models

- Feature learning is absent in random feature models. How to go beyond random feature models?

Random Feature Models

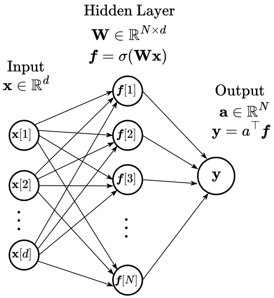
- Feature learning is absent in random feature models. How to go beyond random feature models?
- Realistic training is beyond reach (for now). Existing theoretical approaches:
  - Tensor programs. (Greg Yang et al., 2021+)
  - One step of gradient descent on first layer weights. Damian et al. (2022), Ba et al. (2022), Dandi et al. (2023), Cui et al. (2024), ...
  - ...

One Gradient Step

One Gradient Step Update

We train the network as follows:

Damian et al. (2022), Ba et al. (2022), Dandi et al. (2023), Cui et al. (2024), ...



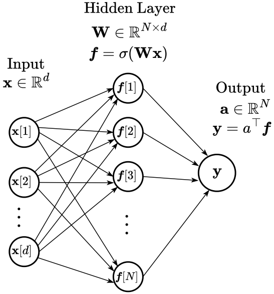
One Gradient Step Update

We train the network as follows:

Damian et al. (2022), Ba et al. (2022), Dandi et al. (2023), Cui et al. (2024), ...

1. Initialize

$$\boldsymbol{a} \sim \mathcal{N}\left(\mathbf{0}_N, \frac{1}{N}\mathbf{I}_N\right), \quad \text{and} \quad [\mathbf{W}_0]_{ij} \sim \mathcal{N}\left(0, \frac{1}{d}\right)$$



One Gradient Step Update

We train the network as follows:

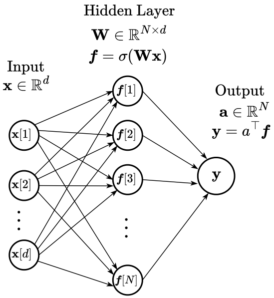
Damian et al. (2022), Ba et al. (2022), Dandi et al. (2023), Cui et al. (2024), ...

1. Initialize

$$\boldsymbol{a} \sim \mathcal{N}\left(\mathbf{0}_N, \frac{1}{N}\mathbf{I}_N\right), \quad \text{and} \quad [\mathbf{W}_0]_{ij} \sim \mathcal{N}\left(0, \frac{1}{d}\right)$$

2. Take one gradient step on the empirical MSE loss $\mathcal{L}$:

$$\mathbf{W} = \mathbf{W}_0 - \eta \frac{\partial}{\partial \mathbf{W}} \left(\frac{1}{2n} \|\boldsymbol{y} - \sigma(\mathbf{X}\mathbf{W}^\top)\boldsymbol{a}\|_2^2 \right) \Big|_{\mathbf{W}_0, \boldsymbol{a}}$$



One Gradient Step Update

We train the network as follows:

Damian et al. (2022), Ba et al. (2022), Dandi et al. (2023), Cui et al. (2024), ...

1. Initialize

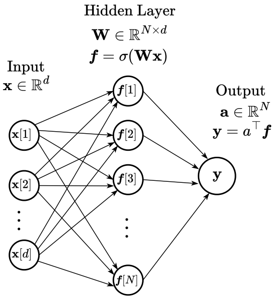
$$\boldsymbol{a} \sim \mathcal{N}\left(\mathbf{0}_N, \frac{1}{N}\mathbf{I}_N\right), \quad \text{and} \quad [\mathbf{W}_0]_{ij} \sim \mathcal{N}\left(0, \frac{1}{d}\right)$$

2. Take one gradient step on the empirical MSE loss $\mathcal{L}$:

$$\mathbf{W} = \mathbf{W}_0 - \eta \frac{\partial}{\partial \mathbf{W}}\left(\frac{1}{2n}\|\boldsymbol{y} - \sigma(\mathbf{X}\mathbf{W}^\top)\boldsymbol{a}\|_2^2\right)\Big|_{\mathbf{W}_0, \boldsymbol{a}}$$

3. Fit $\boldsymbol{a}$ via ridge regression on independent dataset $\tilde{\mathbf{X}}, \tilde{\boldsymbol{y}}$ of same size:

$$\hat{\boldsymbol{a}} = \underset{\tilde{\boldsymbol{a}} \in \mathbb{R}^N}{\arg\min} \frac{1}{n}\|\tilde{\boldsymbol{y}} - \mathbf{F}\tilde{\boldsymbol{a}}\|_2^2 + \lambda\|\tilde{\boldsymbol{a}}\|_2^2, \quad \mathbf{F} = \sigma(\tilde{\mathbf{X}}\mathbf{W}^\top) \in \mathbb{R}^{n \times N}.$$



Data Generation

- **Data generation:**

$$x_i \overset{i.i.d.}{\sim} \mathsf{N}(0, \mathbf{I}_d), \quad y_i = f_\star(\boldsymbol{x}_i) + \varepsilon_i,$$

where $\varepsilon_i \overset{i.i.d.}{\sim} \mathsf{N}(0, \sigma_\varepsilon^2)$. Let $\mathbf{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]^\top \in \mathbb{R}^{n \times d}$, $\boldsymbol{y} = (y_1, \ldots, y_n)^\top \in \mathbb{R}^n$ and $\tilde{\mathbf{X}} = [\boldsymbol{x}_{n+1}, \ldots, \boldsymbol{x}_{2n}]^\top \in \mathbb{R}^{n \times d}$, $\tilde{\boldsymbol{y}} = (y_{n+1}, \ldots, y_{2n})^\top \in \mathbb{R}^n$.

Data Generation

- **Data generation:**

$$x_i \overset{i.i.d.}{\sim} \mathsf{N}(0, \mathbf{I}_d), \quad y_i = f_\star(\boldsymbol{x}_i) + \varepsilon_i,$$

  where $\varepsilon_i \overset{i.i.d.}{\sim} \mathsf{N}(0, \sigma_\varepsilon^2)$. Let $\mathbf{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]^\top \in \mathbb{R}^{n \times d}$, $\boldsymbol{y} = (y_1, \ldots, y_n)^\top \in \mathbb{R}^n$ and $\tilde{\mathbf{X}} = [\boldsymbol{x}_{n+1}, \ldots, \boldsymbol{x}_{2n}]^\top \in \mathbb{R}^{n \times d}$, $\tilde{\boldsymbol{y}} = (y_{n+1}, \ldots, y_{2n})^\top \in \mathbb{R}^n$.
- With one step, only a single-index approximation can be learned. (see e.g., Dandi et al. (2023), etc.) Thus, we let

$$f_\star(\boldsymbol{x}_i) = \sigma_\star(\boldsymbol{\beta}_\star^\top \boldsymbol{x}_i)$$

Prior Work

Partial understanding in prior work:

- Ba et al. (2022) show that if $\eta = O(1)$, still no nonlinear component of the teacher function can be learned. Performance is still worse than linear regression on full data.

Prior Work

Partial understanding in prior work:

- Ba et al. (2022) show that if $\eta = O(1)$, still no nonlinear component of the teacher function can be learned. Performance is still worse than linear regression on full data.

- However, with $\eta = O(\sqrt{n})$ the one-step updated random features model can outperform linear and kernel predictors.

Prior Work

Partial understanding in prior work:

- Ba et al. (2022) show that if $\eta = O(1)$, still no nonlinear component of the teacher function can be learned. Performance is still worse than linear regression on full data.

- However, with $\eta = O(\sqrt{n})$ the one-step updated random features model can outperform linear and kernel predictors.

- How? What features are learned? By how much does performance improve?

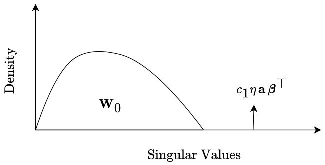
Spectral Analysis of the Feature Matrix

Spectrum of the Weights

$$\mathbf{W} = \mathbf{W}_0 - \eta \frac{\partial}{\partial \mathbf{W}} \left(\frac{1}{2n} \|\boldsymbol{y} - \sigma(\mathbf{X}\mathbf{W}^\top)\boldsymbol{a}\|_2^2 \right) \Big|_{\mathbf{W}_0, \boldsymbol{a}}$$

$$\approx \mathbf{W}_0 + \eta c_1 \boldsymbol{a} \left(\frac{\mathbf{X}^\top \boldsymbol{y}}{n} \right)^\top$$

Ba et al. (2022)

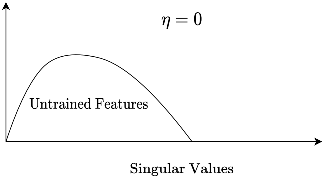


The vector $\boldsymbol{\beta} := \frac{\mathbf{X}^\top \boldsymbol{y}}{n}$ is aligned with $\boldsymbol{\beta}_\star$.

Spectrum of the Feature Matrix

Updated Weight Matrix: $\mathbf{W} \approx \mathbf{W}_0 + \eta c_1 \boldsymbol{a} \boldsymbol{\beta}^\top$

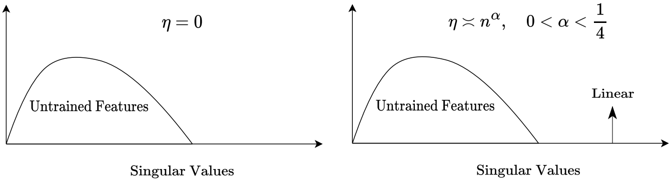
Feature Matrix: $\mathbf{F} = \sigma(\tilde{\mathbf{X}}\mathbf{W}^\top) \approx \sigma(\tilde{\mathbf{X}}\mathbf{W}_0^\top + c_1 \eta \tilde{\mathbf{X}} \boldsymbol{\beta} \boldsymbol{a}^\top) \in \mathbb{R}^{n \times N}$



Spectrum of the Feature Matrix

Updated Weight Matrix: $\mathbf{W} \approx \mathbf{W}_0 + \eta c_1 \boldsymbol{a}\boldsymbol{\beta}^\top$

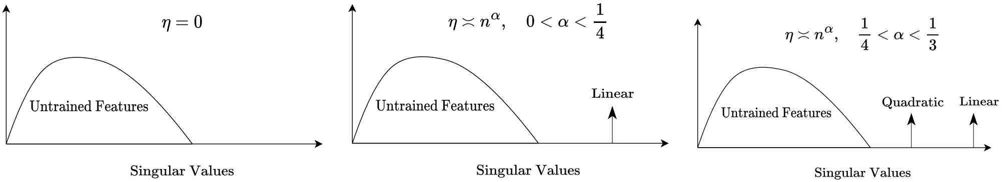
Feature Matrix: $\mathbf{F} = \sigma(\tilde{\mathbf{X}}\mathbf{W}^\top) \approx \sigma(\tilde{\mathbf{X}}\mathbf{W}_0^\top + c_1\eta\tilde{\mathbf{X}}\boldsymbol{\beta}\boldsymbol{a}^\top) \in \mathbb{R}^{n\times N}$



Spectrum of the Feature Matrix

Updated Weight Matrix: $\mathbf{W} \approx \mathbf{W}_0 + \eta c_1 \boldsymbol{a}\boldsymbol{\beta}^\top$

Feature Matrix: $\mathbf{F} = \sigma(\tilde{\mathbf{X}}\mathbf{W}^\top) \approx \sigma(\tilde{\mathbf{X}}\mathbf{W}_0^\top + c_1\eta\tilde{\mathbf{X}}\boldsymbol{\beta}\boldsymbol{a}^\top) \in \mathbb{R}^{n\times N}$



Non-linearity pushes spikes outside the spectrum of random features $\sigma(\tilde{\mathbf{X}}\mathbf{W}_0^\top)$, creating non-linear features.

Spectrum of Updated Feature Matrix

Recall $\mathbf{F}_0 = \sigma(\tilde{\mathbf{X}}\mathbf{W}_0^\top)$, $\mathbf{F} = \sigma(\tilde{\mathbf{X}}\mathbf{W}^\top) \approx \sigma(\tilde{\mathbf{X}}\mathbf{W}_0^\top + c_1\eta\tilde{\mathbf{X}}\boldsymbol{\beta}\boldsymbol{a}^\top)$. Hermite expansion in L^2 of activation function $\sigma : \mathbb{R} \to \mathbb{R}$, $c_1 \neq 0$:

$$\sigma(z) = \sum_{k=1}^{\infty} c_k H_k(z), \quad c_k = \frac{1}{k!}\mathbb{E}_{Z\sim\mathsf{N}(0,1)}[\sigma(Z)H_k(Z)].$$

Theorem

Let $\eta \asymp n^\alpha$ with $\frac{\ell-1}{2\ell} < \alpha < \frac{\ell}{2\ell+2}$ for some $\ell \in \mathbb{N}$. Under conditions,

$$\|\mathbf{F} - \mathbf{F}_\ell\|_{\mathrm{op}} = o_P(\sqrt{n}), \text{ with } \mathbf{F}_\ell := \mathbf{F}_0 + \sum_{k=1}^{\ell} c_1^k c_k \eta^k (\tilde{\mathbf{X}}\boldsymbol{\beta})^{\circ k}(\boldsymbol{a}^{\circ k})^\top.$$

Spectrum of Updated Feature Matrix

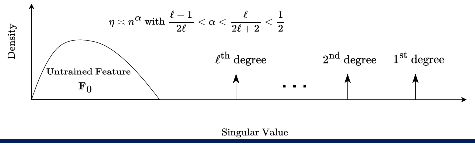
Recall $\mathbf{F}_0 = \sigma(\tilde{\mathbf{X}}\mathbf{W}_0^\top)$, $\mathbf{F} = \sigma(\tilde{\mathbf{X}}\mathbf{W}^\top) \approx \sigma(\tilde{\mathbf{X}}\mathbf{W}_0^\top + c_1\eta\tilde{\mathbf{X}}\boldsymbol{\beta}\boldsymbol{a}^\top)$. Hermite expansion in L^2 of activation function $\sigma : \mathbb{R} \to \mathbb{R}$, $c_1 \neq 0$:

$$\sigma(z) = \sum_{k=1}^{\infty} c_k H_k(z), \quad c_k = \frac{1}{k!}\mathbb{E}_{Z\sim\mathsf{N}(0,1)}[\sigma(Z)H_k(Z)].$$

Theorem

Let $\eta \asymp n^\alpha$ with $\frac{\ell-1}{2\ell} < \alpha < \frac{\ell}{2\ell+2}$ for some $\ell \in \mathbb{N}$. Under conditions,

$$\|\mathbf{F} - \mathbf{F}_\ell\|_{\mathrm{op}} = o_P(\sqrt{n}), \text{ with } \mathbf{F}_\ell := \mathbf{F}_0 + \sum_{k=1}^{\ell} c_1^k c_k \eta^k (\tilde{\mathbf{X}}\boldsymbol{\beta})^{\circ k}(\boldsymbol{a}^{\circ k})^\top.$$



Some Intuition

- Recall

$$\|\mathbf{F} - \mathbf{F}_\ell\|_{\mathrm{op}} = o(\sqrt{n}),\ \text{with } \mathbf{F}_\ell := \mathbf{F}_0 + \sum_{k=1}^{\ell} c_1^k c_k \eta^k (\tilde{\mathbf{X}}\boldsymbol{\beta})^{\circ k} (\boldsymbol{a}^{\circ k})^\top.$$

- Consider $c_1^k c_k \eta^k (\tilde{\mathbf{X}}\boldsymbol{\beta})^{\circ k} (\boldsymbol{a}^{\circ k})^\top$. Its operator norm is of order

$$\eta^k \|(\tilde{\mathbf{X}}\boldsymbol{\beta})^{\circ k}\| \|\boldsymbol{a}^{\circ k}\| \approx n^{\alpha k} n^{1/2} n^{1/2-k/2} = n^{(\alpha-1/2)k+1}$$

Some Intuition

- Recall

$$\|\mathbf{F} - \mathbf{F}_\ell\|_{\mathrm{op}} = o(\sqrt{n}), \text{ with } \mathbf{F}_\ell := \mathbf{F}_0 + \sum_{k=1}^{\ell} c_1^k c_k \eta^k (\tilde{\mathbf{X}}\boldsymbol{\beta})^{\circ k} (\boldsymbol{a}^{\circ k})^\top.$$

- Consider $c_1^k c_k \eta^k (\tilde{\mathbf{X}}\boldsymbol{\beta})^{\circ k} (\boldsymbol{a}^{\circ k})^\top$. Its operator norm is of order

$$\eta^k \|(\tilde{\mathbf{X}}\boldsymbol{\beta})^{\circ k}\| \|\boldsymbol{a}^{\circ k}\| \approx n^{\alpha k} n^{1/2} n^{1/2-k/2} = n^{(\alpha-1/2)k+1}$$

- We have $n^{(\alpha-1/2)k+1} \gg n^{1/2}$ iff $(1/2-\alpha)k < 1/2$ iff $\alpha > 1/2 - 1/(2k) = (k-1)/(2k)$

Training/Test Error

Universality

- What error does the trained neural network achieve?

- To find this, we show that the limiting behavior of test/train error is unchanged if

Universality

- What error does the trained neural network achieve?

- To find this, we show that the limiting behavior of test/train error is unchanged if

    1. we replace $\mathbf{F} = \sigma(\tilde{\mathbf{X}}\mathbf{W}^\top)$ with

$$\mathbf{F} = \mathbf{F}_0 + \sum_{k=1}^{\ell} c_1^k c_k \eta^k (\tilde{\mathbf{X}}\boldsymbol{\beta})^{\circ k} (\boldsymbol{a}^{\circ k})^\top.$$

Universality

- What error does the trained neural network achieve?

- To find this, we show that the limiting behavior of test/train error is <u>unchanged</u> if

  1. we replace $\mathbf{F} = \sigma(\tilde{\mathbf{X}}\mathbf{W}^\top)$ with
  $$\mathbf{F} = \mathbf{F}_0 + \sum_{k=1}^{\ell} c_1^k c_k \eta^k (\tilde{\mathbf{X}}\boldsymbol{\beta})^{\circ k} (\boldsymbol{a}^{\circ k})^\top.$$
  2. we replace $\mathbf{F}_0$ with $c_1 \tilde{\mathbf{X}}\mathbf{W}_0^\top + c_{>1}\mathbf{Z}$, $c_{>1} = (\sum_{k=2}^{\infty} k! c_k^2)^{1/2}$. (Gaussian equivalence; needs work here)

Training/Test Errors

Theorem

Let $\ell \in \mathbb{N}$ *and* $\eta \asymp n^{\alpha}$ *with* $\frac{\ell-1}{2\ell} < \alpha < \frac{\ell}{2\ell+2}$, *then under conditions, for the learned feature map* $\mathbf{F}$ *and the untrained feature map* $\mathbf{F}_0$, *we have*

$$\mathcal{L}_{\text{tr}}(\mathbf{F}_0) - \mathcal{L}_{\text{tr}}(\mathbf{F}) \to_P \Delta_{\text{tr}} \geq 0,$$

$$\mathcal{L}_{\text{te}}(\mathbf{F}_0) - \mathcal{L}_{\text{te}}(\mathbf{F}) \to_P \Delta_{\text{te}} \geq 0.$$

For test error, cover $\ell = 1, 2$.

Training/Test Errors

Theorem

Let $\ell \in \mathbb{N}$ and $\eta \asymp n^{\alpha}$ with $\frac{\ell-1}{2\ell} < \alpha < \frac{\ell}{2\ell+2}$, then under conditions, for the learned feature map $\mathbf{F}$ and the untrained feature map $\mathbf{F}_0$, we have

$$\mathcal{L}_{\mathrm{tr}}(\mathbf{F}_0) - \mathcal{L}_{\mathrm{tr}}(\mathbf{F}) \to_P \Delta_{\mathrm{tr}} \geq 0,$$

$$\mathcal{L}_{\mathrm{te}}(\mathbf{F}_0) - \mathcal{L}_{\mathrm{te}}(\mathbf{F}) \to_P \Delta_{\mathrm{te}} \geq 0.$$

For test error, cover $\ell = 1, 2$.

Recent breakthrough: Test error for $\alpha = 1/$ via equivalent *spiked random features* model. Cui et al. (2024)

Details of Training/Test Error Analysis

Conditions:

1. We let $f_\star : \mathbb{R}^d \to \mathbb{R}$ be $f_\star(\boldsymbol{x}) = \sigma_\star(\boldsymbol{x}^\top \boldsymbol{\beta}_\star)$ for all $\boldsymbol{x}$, where $\boldsymbol{\beta}_\star \in \mathbb{R}^d$ is an unknown parameter with $\boldsymbol{\beta}_\star \sim \mathsf{N}(0, \frac{1}{d}\mathbf{I}_d)$ and $\sigma_\star : \mathbb{R} \to \mathbb{R}$ is a $\Theta(1)$-Lipschitz *teacher activation* function.
2. The teacher activation $\sigma_\star : \mathbb{R} \to \mathbb{R}$ has the following Hermite expansion in L^2:

$$\sigma_\star(z) = \sum_{k=1}^{\infty} c_{\star,k} H_k(z), \;\; c_{\star,k} = \frac{1}{k!} \mathbb{E}_{Z \sim \mathsf{N}(0,1)}[\sigma_\star(Z) H_k(Z)].$$

Also, we define $c_\star = (\sum_{k=1}^{\infty} k! c_{\star,k}^2)^{\frac{1}{2}}$.

Details of Training/Test Error Analysis

Conditions, ctd.:

1. The activation function $\sigma : \mathbb{R} \to \mathbb{R}$ has the following Hermite expansion in L^2:

$$\sigma(z) = \sum_{k=1}^{\infty} c_k H_k(z), \quad c_k = \frac{1}{k!}\mathbb{E}_{Z\sim \mathsf{N}(0,1)}[\sigma(Z)H_k(Z)].$$

The coefficients satisfy $c_1 \neq 0$ and $c_k^2 k! \leq Ck^{-\frac{3}{2}-\omega}$ for some $C, \omega > 0$ and for all $k \geq 1$. Moreover, the first three derivatives of σ exist almost surely, and are bounded.

Special Case of Training Loss

Recall $n, d, N \to \infty$ with $d/n \to \phi$ and $d/N \to \psi$; c_i are Hermite coeffs of fitted RF nonlin. σ; $c_{\star,i}$ are Hermite coeffs of true nonlin. $\sigma_\star$

Proposition (Learning Linear & Quadratic Features)

If $c_1 \neq 0$ and $\eta \asymp n^\alpha$ with $0 < \alpha < \frac{1}{4}$, we have

$$\mathcal{L}_{\mathrm{tr}}(\mathbf{F}_0) - \mathcal{L}_{\mathrm{tr}}(\mathbf{F}) \to_P \Delta_1 := \frac{\psi \lambda c_{\star,1}^4 m_2}{\phi[c_{\star,1}^2 + \phi(c_\star^2 + \sigma_\varepsilon^2)]} \geq 0. \tag{1}$$

If also $c_2 \neq 0$ and $\eta \asymp n^\alpha$ with $\frac{1}{4} < \alpha < \frac{1}{3}$, we have

$$\mathcal{L}_{\mathrm{tr}}(\mathbf{F}_0) - \mathcal{L}_{\mathrm{tr}}(\mathbf{F}) \to_P \Delta_2 := \Delta_1 + \frac{4\psi \lambda c_{\star,1}^4 c_{\star,2}^2 m_1}{3\phi[\phi(c_\star^2 + \sigma_\varepsilon^2) + c_{\star,1}^2]^2} \geq 0. \tag{2}$$

Analysis of Training/Test Error

- **Limiting traces:** For $\mathbf{F}_0 = \sigma(\tilde{\mathbf{X}}_0 \mathbf{W}^\top)$, (Pennington and Worah, 2017; Adlam and Pennington, 2020)

$$m_1 := \frac{\phi}{\psi} \lim_{d,n,N\to\infty} \operatorname{tr}((\mathbf{F}_0\mathbf{F}_0^\top + \lambda n \mathbf{I}_n)^{-1}) > 0$$
$$m_2 := \frac{\phi}{\psi} \lim_{d,n,N\to\infty} \frac{1}{d} \operatorname{tr}(\tilde{\mathbf{X}}^\top (\mathbf{F}_0\mathbf{F}_0^\top + \lambda n \mathbf{I}_n)^{-1} \tilde{\mathbf{X}}) > 0.$$

- Unique solutions of the system of equations:

$$\phi\,(m_1 - m_2)\left(c_{>1}^2 m_1 + c_1^2 m_2\right) + \Psi(m_1, m_2) = 0,$$
$$\frac{\phi}{\psi}\left(c_1^2 m_1 m_2 + \phi\,(m_2 - m_1)\right) + \Psi(m_1, m_2) = 0,$$

where $\Psi(m_1, m_2) = c_1^2 m_1 m_2\,(\lambda \psi m_1/\phi - 1)$ and $c_{>1} = (\sum_{k=2}^{\infty} k! c_k^2)^{1/2}$.

Special Cases of Test Error

Proposition (Learning Linear & Quadratic Features)

If $c_1 \neq 0$ and $\eta \asymp n^\alpha$ with $0 < \alpha < \frac{1}{4}$, we have

$$\mathcal{L}_{\mathrm{te}}(\mathbf{F}_0) - \mathcal{L}_{\mathrm{te}}(\mathbf{F}) \to_P \Lambda_1 := \frac{c_{\star,1}^4}{[c_{\star,1}^2 + \phi(c_\star^2 + \sigma_\varepsilon^2)]} \left(-\frac{\partial m_2}{\partial \lambda}\right) \geq 0. \tag{3}$$

If also $c_2 \neq 0$ and $\eta \asymp n^\alpha$ with $\frac{1}{4} < \alpha < \frac{1}{3}$, we have

$$\mathcal{L}_{\mathrm{te}}(\mathbf{F}_0) - \mathcal{L}_{\mathrm{te}}(\mathbf{F}) \to_P \Lambda_1 + \frac{4c_{\star,1}^4 c_{\star,2}^2}{3[c_{\star,1}^2 + \phi(c_\star^2 + \sigma_\varepsilon^2)]^2 m_1^2} \left(-\frac{\partial m_1}{\partial \lambda}\right) \geq 0. \tag{4}$$

Training Error: General Case

Let $\xi_{i,j}$, $i,j \in \{0,1,\ldots\}$ s.t. for any $p \in \mathbb{N}$ and $x \in \mathbb{R}$, $x^p = \sum_{i=0}^{p} \xi_{p,i} H_i(x)$.

Theorem

Let $\ell \in \mathbb{N}$. If $c_1, \cdots, c_\ell \neq 0$, and $\eta \asymp n^\alpha$ with $\frac{\ell-1}{2\ell} < \alpha < \frac{\ell}{2\ell+2}$, then for the learned feature map $\mathbf{F}$ and the untrained feature map $\mathbf{F}_0$, we have $\mathcal{L}_{\mathrm{tr}}(\mathbf{F}_0) - \mathcal{L}_{\mathrm{tr}}(\mathbf{F}) \to_P \Delta_\ell \geq 0$, where

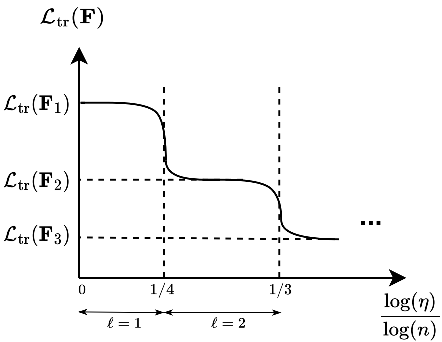
$$\Delta_\ell = \lambda \sum_{p=1}^{\ell} \sum_{q=1}^{\ell} c_{\star,p} c_{\star,q} r_p r_q \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \Omega_{i,j} \left(\phi(c_\star^2 + \sigma_\varepsilon^2) + c_{\star,1}^2 \right)^{(i+j)/2} \xi_{i,p} \xi_{j,q},$$

in which Ω is an invertible matrix with, $\forall i,j \in [\ell]$,

$$[\Omega^{-1}]_{i,j} = \left(c_{\star,1}^2 + \phi(c_\star^2 + \sigma_\varepsilon^2) \right)^{(i+j)/2} \frac{\psi}{\phi} \left[m_2 \xi_{i,1} \xi_{j,1} + m_1 \sum_{k=0,\ k\neq 1}^{\min(i,j)} k!\, \xi_{i,k} \xi_{j,k} \right],$$

and $r_p = \frac{p!\psi m_1}{\phi} \left(\frac{c_{\star,1}}{\sqrt{\phi(c_\star^2+\sigma_\varepsilon^2)+c_{\star,1}^2}} \right)^p, p \neq 1;\ r_p = \frac{\psi m_2}{\phi} \frac{c_{\star,1}}{\sqrt{\phi(c_\star^2+\sigma_\varepsilon^2)+c_{\star,1}^2}}, p = 1.$

Training Errors: “Staircase property"



Fit increasingly larger set of polynomial features. Consistent with *staircase property*. (Abbe et al., 2021, 2022; Berthier et al., 2023), . . .

Elements of the Proofs

- Training loss: $\mathcal{L}_{\rm tr}(\mathbf{F}) = \lambda \tilde{\boldsymbol{y}}^\top (\mathbf{F}\mathbf{F}^\top + \lambda n \mathbf{I}_n)^{-1} \tilde{\boldsymbol{y}}$.
- **Equivalence Theorems**: Replace $\mathbf{F}$ with the spiked approximation.

Elements of the Proofs

- Training loss: $\mathcal{L}_{\text{tr}}(\mathbf{F}) = \lambda \tilde{\boldsymbol{y}}^\top (\mathbf{F}\mathbf{F}^\top + \lambda n \mathbf{I}_n)^{-1} \tilde{\boldsymbol{y}}$.
- **Equivalence Theorems**: Replace $\mathbf{F}$ with the spiked approximation.
- **Woodbury Formula**: express the training/test errors in terms of $\bar{\mathbf{R}}_0 = (\mathbf{F}_0\mathbf{F}_0^\top + \lambda n \mathbf{I}_n)^{-1}$ and the non-linear spikes.

Elements of the Proofs

- Training loss: $\mathcal{L}_{\mathrm{tr}}(\mathbf{F}) = \lambda \tilde{\boldsymbol{y}}^\top (\mathbf{F}\mathbf{F}^\top + \lambda n \mathbf{I}_n)^{-1} \tilde{\boldsymbol{y}}$.
- **Equivalence Theorems**: Replace $\mathbf{F}$ with the spiked approximation.
- **Woodbury Formula**: express the training/test errors in terms of $\bar{\mathbf{R}}_0 = (\mathbf{F}_0\mathbf{F}_0^\top + \lambda n \mathbf{I}_n)^{-1}$ and the non-linear spikes.
- **Nonlinear Terms:** Find limits of $v_1^\top (\mathbf{F}_0\mathbf{F}_0^\top + \lambda n \mathbf{I}_n)^{-1} v_2$, for $v_1, v_2 \in \{H_q(\tilde{\mathbf{X}}\boldsymbol{\beta}), H_q(\tilde{\mathbf{X}}\boldsymbol{\beta}_\star), \mathbf{F}_0\boldsymbol{a}, \mathbf{F}_0\boldsymbol{a}^{\circ 2}\}$.

Elements of the Proofs

- Training loss: $\mathcal{L}_{\text{tr}}(\mathbf{F}) = \lambda \tilde{\boldsymbol{y}}^\top (\mathbf{F}\mathbf{F}^\top + \lambda n \mathbf{I}_n)^{-1} \tilde{\boldsymbol{y}}$.
- **Equivalence Theorems**: Replace $\mathbf{F}$ with the spiked approximation.
- **Woodbury Formula**: express the training/test errors in terms of $\bar{\mathbf{R}}_0 = (\mathbf{F}_0\mathbf{F}_0^\top + \lambda n \mathbf{I}_n)^{-1}$ and the non-linear spikes.
- **Nonlinear Terms:** Find limits of $v_1^\top (\mathbf{F}_0\mathbf{F}_0^\top + \lambda n \mathbf{I}_n)^{-1} v_2$, for $v_1, v_2 \in \{H_q(\tilde{\mathbf{X}}\boldsymbol{\beta}), H_q(\tilde{\mathbf{X}}\boldsymbol{\beta}_\star), \mathbf{F}_0\boldsymbol{a}, \mathbf{F}_0\boldsymbol{a}^{\circ 2}\}$.
- **Gaussian Equivalence:**

$$\mathbf{F}_0 = \sigma(\tilde{\mathbf{X}}\mathbf{W}_0^\top) \leftarrow c_1 \tilde{\mathbf{X}}\mathbf{W}_0^\top + c_{>1}\mathbf{Z}, \quad Z_{ij} \overset{i.i.d.}{\sim} \mathsf{N}(0,1).$$

The interaction between the first ℓ Hermite components of $\tilde{\boldsymbol{y}}$ and the spike terms is non-vanishing.

Elements of the Proofs

- Training loss: $\mathcal{L}_{\mathrm{tr}}(\mathbf{F}) = \lambda \tilde{\boldsymbol{y}}^\top (\mathbf{F}\mathbf{F}^\top + \lambda n \mathbf{I}_n)^{-1} \tilde{\boldsymbol{y}}$.
- **Equivalence Theorems**: Replace $\mathbf{F}$ with the spiked approximation.
- **Woodbury Formula**: express the training/test errors in terms of $\bar{\mathbf{R}}_0 = (\mathbf{F}_0\mathbf{F}_0^\top + \lambda n \mathbf{I}_n)^{-1}$ and the non-linear spikes.
- **Nonlinear Terms:** Find limits of $v_1^\top (\mathbf{F}_0\mathbf{F}_0^\top + \lambda n \mathbf{I}_n)^{-1} v_2$, for $v_1, v_2 \in \{H_q(\tilde{\mathbf{X}}\boldsymbol{\beta}), H_q(\tilde{\mathbf{X}}\boldsymbol{\beta}_\star), \mathbf{F}_0\boldsymbol{a}, \mathbf{F}_0\boldsymbol{a}^{\circ 2}\}$.
- **Gaussian Equivalence:**

$$\mathbf{F}_0 = \sigma(\tilde{\mathbf{X}}\mathbf{W}_0^\top) \leftarrow c_1\tilde{\mathbf{X}}\mathbf{W}_0^\top + c_{>1}\mathbf{Z}, \quad Z_{ij} \overset{i.i.d.}{\sim} \mathsf{N}(0,1).$$

  The interaction between the first ℓ Hermite components of $\tilde{\boldsymbol{y}}$ and the spike terms is non-vanishing.
- Concentration + Adlam and Pennington (2020): find limits in terms of m_1, m_2.

Elements of the Proofs

- **Gaussian Equivalence:** Need to show that can replace

$$\mathbf{F}_0 = \sigma(\tilde{\mathbf{X}}\mathbf{W}_0^\top) \leftarrow c_1\tilde{\mathbf{X}}\mathbf{W}_0^\top + c_{>1}\mathbf{Z}, \quad Z_{ij} \overset{i.i.d.}{\sim} \mathsf{N}(0,1)$$

  w/o changing the limit of the nonlinear terms.
- We could not deduce this from existing results
- Adopt Lindeberg exchange method + concentration of QF + spectrum of kernel random matrices (El Karoui, 2010)

Simulations

Simulation Results

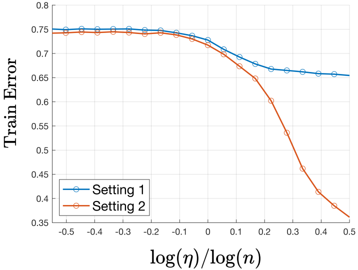
We consider

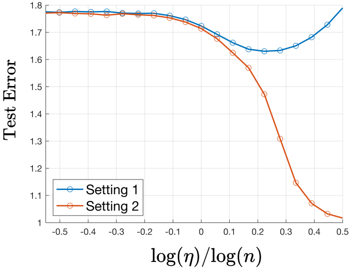
$$\textbf{Setting 1}: y = H_1(\boldsymbol{\beta}_\star^\top \boldsymbol{x}) + \varepsilon, \quad \varepsilon \sim \mathsf{N}(0,1),$$
$$\textbf{Setting 2}: y = H_1(\boldsymbol{\beta}_\star^\top \boldsymbol{x}) + \frac{1}{\sqrt{2}} H_2(\boldsymbol{\beta}_\star^\top \boldsymbol{x}).$$

Simulation Results

We consider

$$\textbf{Setting 1}: y = H_1(\boldsymbol{\beta}_\star^\top \boldsymbol{x}) + \varepsilon, \quad \varepsilon \sim \mathsf{N}(0,1),$$
$$\textbf{Setting 2}: y = H_1(\boldsymbol{\beta}_\star^\top \boldsymbol{x}) + \frac{1}{\sqrt{2}} H_2(\boldsymbol{\beta}_\star^\top \boldsymbol{x}).$$





Conclusion

Conclusion

- One step of gradient descent with step size $\eta \asymp n^{\alpha}$ can lead to *non-linear feature learning* (unlike for the RF model).

Conclusion

- One step of gradient descent with step size $\eta \asymp n^\alpha$ can lead to *non-linear feature learning* (unlike for the RF model).

- Learned features depend on the range of α.

Conclusion

- One step of gradient descent with step size $\eta \asymp n^{\alpha}$ can lead to *non-linear feature learning* (unlike for the RF model).

- Learned features depend on the range of α.

- Thanks! Questions?

NSF
SIMONS FOUNDATION
ALFRED P. SLOAN FOUNDATION
Office of Naval Research Science & Technology
Army Research Office
Air Force Office of Scientific Research

References

References I

Abbe, E., Adsera, E. B., and Misiakiewicz, T. (2022). The merged-staircase property: a necessary and nearly sufficient condition for sgd learning of sparse functions on two-layer neural networks. In *Conference on Learning Theory*, pages 4782–4887. PMLR.

Abbe, E., Boix-Adsera, E., Brennan, M. S., Bresler, G., and Nagaraj, D. (2021). The staircase property: How hierarchical structure can guide deep learning. In *Advances in Neural Information Processing Systems*, pages 26989–27002.

Adlam, B. and Pennington, J. (2020). The neural tangent kernel in high dimensions: Triple descent and a multi-scale theory of generalization. In *International Conference on Machine Learning*.

Ba, J., Erdogdu, M. A., Suzuki, T., Wang, Z., Wu, D., and Yang, G. (2022). High-dimensional asymptotics of feature learning: How one gradient step improves the representation. In *Advances in Neural Information Processing Systems*.

Berthier, R., Montanari, A., and Zhou, K. (2023). Learning time-scales in two-layers neural networks. *arXiv preprint arXiv:2303.00055*.

Bombari, S., Kiyani, S., and Mondelli, M. (2023). Beyond the universal law of robustness: Sharper laws for random features and neural tangent kernels. In *International Conference on Machine Learning*.

Bombari, S. and Mondelli, M. (2023). Stability, generalization and privacy: Precise analysis for random and NTK features. *arXiv preprint arXiv:2305.12100*.

Cheng, X. and Singer, A. (2013). The spectrum of random inner-product kernel matrices. *Random Matrices: Theory and Applications*, 2(04):1350010.

Clarté, L., Loureiro, B., Krzakala, F., and Zdeborová, L. (2023). On double-descent in uncertainty quantification in overparametrized models. In *International Conference on Artificial Intelligence and Statistics*.

Cui, H., Pesce, L., Dandi, Y., Krzakala, F., Lu, Y. M., Zdeborová, L., and Loureiro, B. (2024). Asymptotics of feature learning in two-layer networks after one gradient-step. *arXiv preprint arXiv:2402.04980*.

Damian, A., Lee, J., and Soltanolkotabi, M. (2022). Neural networks can learn representations with gradient descent. In *Conference on Learning Theory*.

Dandi, Y., Krzakala, F., Loureiro, B., Pesce, L., and Stephan, L. (2023). Learning two-layer neural networks, one (giant) step at a time. *arXiv preprint arXiv:2305.18270*.

El Karoui, N. (2010). The spectrum of kernel random matrices. *The Annals of Statistics*, 38(1):1–50.

Fan, Z. and Montanari, A. (2019). The spectral norm of random inner-product kernel matrices. *Probability Theory and Related Fields*, 173(1):27–85.

Gerace, F., Loureiro, B., Krzakala, F., Mézard, M., and Zdeborová, L. (2020). Generalisation error in learning with random features and the hidden manifold model. In *International Conference on Machine Learning*, pages 3452–3462. PMLR.

References II

Ghorbani, B., Mei, S., Misiakiewicz, T., and Montanari, A. (2021). Linearized two-layers neural networks in high dimension. *The Annals of Statistics*, 49(2):1029–1054.

Goldt, S., Loureiro, B., Reeves, G., Krzakala, F., Mézard, M., and Zdeborová, L. (2022). The Gaussian equivalence of generative models for learning with shallow neural networks. In *Mathematical and Scientific Machine Learning*, pages 426–471.

Goldt, S., Mézard, M., Krzakala, F., and Zdeborová, L. (2020). Modeling the influence of data structure on learning in neural networks: The hidden manifold model. *Physical Review X*, 10(4):041044.

Hassani, H. and Javanmard, A. (2022). The curse of overparametrization in adversarial training: Precise analysis of robust generalization for random features regression. *arXiv preprint arXiv:2201.05149*.

Hu, H. and Lu, Y. M. (2023). Universality laws for high-dimensional learning with random features. *IEEE Transactions on Information Theory*, 69(3).

Lee, D., Moniri, B., Huang, X., Dobriban, E., and Hassani, H. (2023). Demystifying disagreement-on-the-line in high dimensions. In *International Conference on Machine Learning*.

Lin, L. and Dobriban, E. (2021). What causes the test error? going beyond bias-variance via ANOVA. *Journal of Machine Learning Research*, 22:155–1.

Mei, S. and Montanari, A. (2022). The generalization error of random features regression: Precise asymptotics and the double descent curve. *Communications on Pure and Applied Mathematics*, 75(4):667–766.

Moniri, B., Lee, D., Hassani, H., and Dobriban, E. (2023). A theory of non-linear feature learning with one gradient step in two-layer neural networks. *arXiv preprint arXiv:2310.07891*.

Pennington, J. and Worah, P. (2017). Nonlinear random matrix theory for deep learning. In *Advances in Neural Information Processing Systems*.

Rahimi, A. and Recht, B. (2007). Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*.