A Modern Take on the Bias-Variance Tradeoff in Neural Networks

Erfan Sobhaei
Sharif University of Technology

First Learning Theory Cirlce

August 2, 2026

1. Introduction

Paper Information

A Modern Take on the Bias-Variance Tradeoff in Neural Networks

Brady Neal **Sarthak Mittal** **Aristide Baratin** **Vinayak Tantia** **Matthew Scicluna**
Simon Lacoste-Julien[†,‡] **Ioannis Mitliagkas**[†]
Mila, Université de Montréal
[†]Canada CIFAR AI Chair [‡]CIFAR Fellow

Motivation

- Too complex models could hurt the performance
- Exceptional empirical success of over-parameterized models
- Relies on practices lacking complete theoretical grounding:
  1. Massive width scale relative to sample size
  2. Over-fitting training data to zero empirical risk
- Need for precise theoretical and empirical characterization

Motivation

- Too complex models could hurt the performance
- Exceptional empirical success of over-parameterized models
- Relies on practices lacking complete theoretical grounding:
  - Massive width scale relative to sample size
  - Over-fitting training data to zero empirical risk
- Need for precise theoretical and empirical characterization

Motivation

- Too complex models could hurt the performance
- Exceptional empirical success of over-parameterized models
- Relies on practices lacking complete theoretical grounding:
  - Massive width scale relative to sample size
  - Over-fitting training data to zero empirical risk
- Need for precise theoretical and empirical characterization

Motivation

- Too complex models could hurt the performance
- Exceptional empirical success of over-parameterized models
- Relies on practices lacking complete theoretical grounding:
  - Massive width scale relative to sample size
  - Over-fitting training data to zero empirical risk
- Need for precise theoretical and empirical characterization

Motivation

- Too complex models could hurt the performance
- Exceptional empirical success of over-parameterized models
- Relies on practices lacking complete theoretical grounding:
  - Massive width scale relative to sample size
  - Over-fitting training data to zero empirical risk
- Need for precise theoretical and empirical characterization

Motivation

- Too complex models could hurt the performance
- Exceptional empirical success of over-parameterized models
- Relies on practices lacking complete theoretical grounding:
  - Massive width scale relative to sample size
  - Over-fitting training data to zero empirical risk
- Need for precise theoretical and empirical characterization

Rethinking Capacity Beyond Parameters

- Modern paradigm shifts focus to alternative mechanisms:
  - Implicit Regularization (Neyshabur et al., 2015)
  - Optimization Landscape (Soudry et al., 2024)
  - Novel Capacity Metrics (Liang et al., 2019)
- Necessity to revisit classical bias-variance

Rethinking Capacity Beyond Parameters

- Modern paradigm shifts focus to alternative mechanisms:
  - Implicit Regularization (Neyshabur et al., 2015)
  - Optimization Landscape (Soudry et al., 2024)
  - Novel Capacity Metrics (Liang et al., 2019)
- Necessity to revisit classical bias-variance

Rethinking Capacity Beyond Parameters

- Modern paradigm shifts focus to alternative mechanisms:
  - Implicit Regularization (Neyshabur et al., 2015)
  - Optimization Landscape (Soudry et al., 2024)
  - Novel Capacity Metrics (Liang et al., 2019)
- Necessity to revisit classical bias-variance

Rethinking Capacity Beyond Parameters

- Modern paradigm shifts focus to alternative mechanisms:
  - Implicit Regularization (Neyshabur et al., 2015)
  - Optimization Landscape (Soudry et al., 2024)
  - Novel Capacity Metrics (Liang et al., 2019)

- Necessity to revisit classical bias-variance

Rethinking Capacity Beyond Parameters

- Modern paradigm shifts focus to alternative mechanisms:
  - Implicit Regularization (Neyshabur et al., 2015)
  - Optimization Landscape (Soudry et al., 2024)
  - Novel Capacity Metrics (Liang et al., 2019)

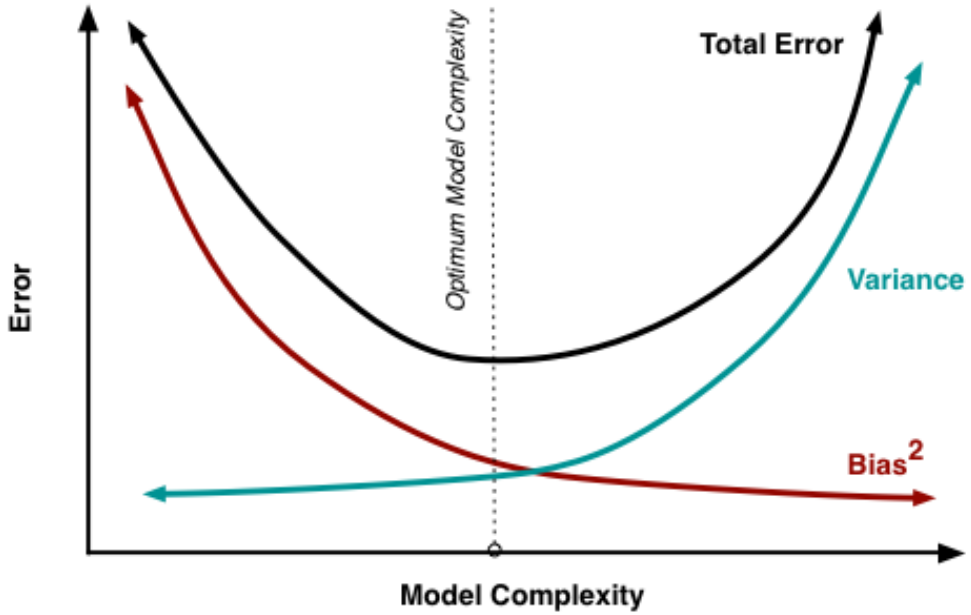
- Necessity to revisit classical bias-variance

The Classical Dogma

- "The price to pay for achieving low bias is high variance." (Geman et al., 1992)

- Classical expectation:

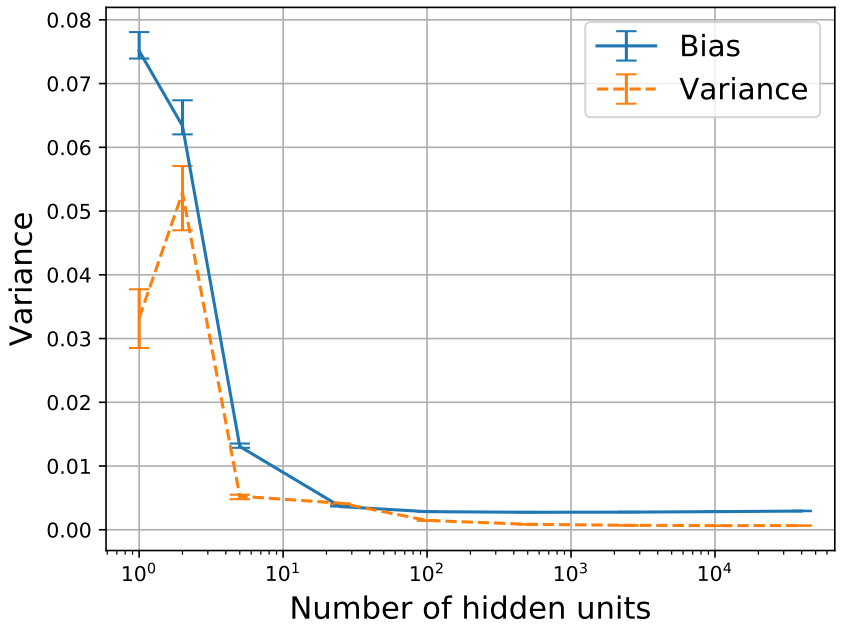
$$\uparrow \text{Model Capacity} \implies \uparrow \text{Variance Explosion}$$



Revisiting Bias and Variance

- Re-evaluating capacity strictly through network width scaling
- Main Finding:

$$\uparrow \text{Network Width} \implies \downarrow \text{Bias, Variance}$$



2. Theoretical Framework

Problem & Model Setup

- Supervised learning task: $x \in \mathcal{X}$, $y \in \mathcal{Y}$ where $(x, y) \sim \mathcal{D}$
- Training dataset $S = \{(x_i, y_i)\}_{i=1}^{m} \sim \mathcal{D}^m$ sampled i.i.d.
- Model $h_S(x) : \mathcal{X} \to \mathcal{Y}$ parameterized by $\theta \in \mathbb{R}^N$ trained on S
- The quality of predictor h quantified by expected error:

$$\mathcal{E}(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\ell(h(x), y)\right]$$

for some loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$

Problem & Model Setup

- Supervised learning task: $x \in \mathcal{X}$, $y \in \mathcal{Y}$ where $(x, y) \sim \mathcal{D}$
- Training dataset $S = \{(x_i, y_i)\}_{i=1}^{m} \sim \mathcal{D}^m$ sampled i.i.d.
- Model $h_S(x) : \mathcal{X} \to \mathcal{Y}$ parameterized by $\theta \in \mathbb{R}^N$ trained on S
- The quality of predictor h quantified by expected error:

$$\mathcal{E}(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\ell(h(x), y)\right]$$

for some loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$

Problem & Model Setup

- Supervised learning task: $x \in \mathcal{X}$, $y \in \mathcal{Y}$ where $(x, y) \sim \mathcal{D}$
- Training dataset $S = \{(x_i, y_i)\}_{i=1}^{m} \sim \mathcal{D}^m$ sampled i.i.d.
- Model $h_S(x) : \mathcal{X} \to \mathcal{Y}$ parameterized by $\theta \in \mathbb{R}^N$ trained on S
- The quality of predictor h quantified by expected error:

$$\mathcal{E}(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\ell(h(x), y)\right]$$

for some loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$

Problem & Model Setup

- Supervised learning task: $x \in \mathcal{X}$, $y \in \mathcal{Y}$ where $(x, y) \sim \mathcal{D}$
- Training dataset $S = \{(x_i, y_i)\}_{i=1}^m \sim \mathcal{D}^m$ sampled i.i.d.
- Model $h_S(x) : \mathcal{X} \to \mathcal{Y}$ parameterized by $\theta \in \mathbb{R}^N$ trained on S
- The quality of predictor h quantified by expected error:

$$\mathcal{E}(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\ell(h(x), y)\right]$$

for some loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$

Sources of Randomness

- Two sources of randomness:
  - Random variable S: Dataset sampling
  - Random variable O: Optimization
- For fixed S and O, learning algorithm choose parameters $\theta = \mathcal{A}(S, O)$
- Encode randomness due to O in $p(\theta \mid S)$

Sources of Randomness

- Two sources of randomness:
  - Random variable S: Dataset sampling
  - Random variable O: Optimization
- For fixed S and O, learning algorithm choose parameters $\theta = \mathcal{A}(S, O)$
- Encode randomness due to O in $p(\theta \mid S)$

Sources of Randomness

- Two sources of randomness:
  - Random variable S: Dataset sampling
  - Random variable O: Optimization
- For fixed S and O, learning algorithm choose parameters $\theta = \mathcal{A}(S, O)$
- Encode randomness due to O in $p(\theta \mid S)$

Sources of Randomness

- Two sources of randomness:
  - Random variable S: Dataset sampling
  - Random variable O: Optimization
- For fixed S and O, learning algorithm choose parameters $\theta = \mathcal{A}(S, O)$
- Encode randomness due to O in $p(\theta \mid S)$

Sources of Randomness

- Two sources of randomness:
  - Random variable S: Dataset sampling
  - Random variable O: Optimization
- For fixed S and O, learning algorithm choose parameters $\theta = \mathcal{A}(S, O)$
- Encode randomness due to O in $p(\theta \mid S)$

Expected True Risk

- Expected error loss over all randomness:

$$\begin{aligned}\mathcal{R}_m &= \mathbb{E}_\theta \left[\mathcal{E}(h_\theta)\right] \\ &= \mathbb{E}_S \mathbb{E}_{\theta \sim p(\cdot|S)} \left[\mathcal{E}(h_\theta)\right] \\ &= \mathbb{E}_S \mathbb{E}_O \left[\mathcal{E}(h_\theta)\right]\end{aligned}$$

Classical Bias-Variance Decomposition

- For squared-loss, we can decompose expected error into:

$$\mathcal{R}_m = \mathcal{E}_{\text{noise}} + \mathcal{E}_{\text{bias}} + \mathcal{E}_{\text{var}}$$

- $\bar{y}(x)$ defined as $\mathbb{E}[y \mid x]$
- We have:

$$\mathcal{E}_{\text{noise}} = \mathbb{E}_{(x,y)}\left[\|y - \bar{y}(x)\|^2\right],$$

$$\mathcal{E}_{\text{bias}} = \mathbb{E}_x\left[\|\mathbb{E}_\theta[h_\theta(x)] - \bar{y}(x)\|^2\right],$$

$$\mathcal{E}_{\text{var}} = \mathbb{E}_x\left[\text{Var}(h_\theta(x))\right],$$

$$\text{Var}(h_\theta(x)) = \mathbb{E}_\theta\left[\|h_\theta(x) - \mathbb{E}_\theta[h_\theta(x)]\|^2\right],$$

- Goal: Further decomposition of $\mathcal{E}_{\text{var}}$ into two terms

Classical Bias-Variance Decomposition

- For squared-loss, we can decompose expected error into:

$$\mathcal{R}_m = \mathcal{E}_{\text{noise}} + \mathcal{E}_{\text{bias}} + \mathcal{E}_{\text{var}}$$

- $\bar{y}(x)$ defined as $\mathbb{E}[y \mid x]$
- We have:

$$\mathcal{E}_{\text{noise}} = \mathbb{E}_{(x,y)}\left[\|y - \bar{y}(x)\|^2\right],$$

$$\mathcal{E}_{\text{bias}} = \mathbb{E}_x\left[\|\mathbb{E}_\theta[h_\theta(x)] - \bar{y}(x)\|^2\right],$$

$$\mathcal{E}_{\text{var}} = \mathbb{E}_x\left[\text{Var}(h_\theta(x))\right],$$

$$\text{Var}(h_\theta(x)) = \mathbb{E}_\theta\left[\|h_\theta(x) - \mathbb{E}_\theta[h_\theta(x)]\|^2\right],$$

- Goal: Further decomposition of $\mathcal{E}_{\text{var}}$ into two terms

Classical Bias-Variance Decomposition

- For squared-loss, we can decompose expected error into:

$$\mathcal{R}_m = \mathcal{E}_{\text{noise}} + \mathcal{E}_{\text{bias}} + \mathcal{E}_{\text{var}}$$

- $\bar{y}(x)$ defined as $\mathbb{E}[y \mid x]$
- We have:

$$\mathcal{E}_{\text{noise}} = \mathbb{E}_{(x,y)}\left[\|y - \bar{y}(x)\|^2\right],$$

$$\mathcal{E}_{\text{bias}} = \mathbb{E}_x\left[\|\mathbb{E}_\theta[h_\theta(x)] - \bar{y}(x)\|^2\right],$$

$$\mathcal{E}_{\text{var}} = \mathbb{E}_x\left[\text{Var}(h_\theta(x))\right],$$

$$\text{Var}(h_\theta(x)) = \mathbb{E}_\theta\left[\|h_\theta(x) - \mathbb{E}_\theta[h_\theta(x)]\|^2\right],$$

- Goal: Further decomposition of $\mathcal{E}_{\text{var}}$ into two terms

Classical Bias-Variance Decomposition

- For squared-loss, we can decompose expected error into:

$$\mathcal{R}_m = \mathcal{E}_{\text{noise}} + \mathcal{E}_{\text{bias}} + \mathcal{E}_{\text{var}}$$

- $\bar{y}(x)$ defined as $\mathbb{E}[y \mid x]$
- We have:

$$\mathcal{E}_{\text{noise}} = \mathbb{E}_{(x,y)} \left[\|y - \bar{y}(x)\|^2 \right],$$

$$\mathcal{E}_{\text{bias}} = \mathbb{E}_x \left[\|\mathbb{E}_\theta[h_\theta(x)] - \bar{y}(x)\|^2 \right],$$

$$\mathcal{E}_{\text{var}} = \mathbb{E}_x \left[\text{Var}(h_\theta(x)) \right],$$

$$\text{Var}(h_\theta(x)) = \mathbb{E}_\theta \left[\|h_\theta(x) - \mathbb{E}_\theta[h_\theta(x)]\|^2 \right],$$

- Goal: Further decomposition of $\mathcal{E}_{\text{var}}$ into two terms

Term 1: Variance due to Sampling

- (Ensemble) Variance due to sampling:

$$\mathrm{Var}_S\left(\mathbb{E}_O\left[h_\theta(x) \mid S\right]\right)$$

- Variance of an ensemble of infinitely many predictors with different optimization randomness

- Run multiple seeds on a fixed training set.

Term 1: Variance due to Sampling

- (Ensemble) Variance due to sampling:

$$\mathrm{Var}_S\left(\mathbb{E}_O\left[h_\theta(x) \mid S\right]\right)$$

- Variance of an ensemble of infinitely many predictors with different optimization randomness

- Run multiple seeds on a fixed training set.

Term 1: Variance due to Sampling

- (Ensemble) Variance due to sampling:

$$\mathrm{Var}_S\left(\mathbb{E}_O\left[h_\theta(x) \mid S\right]\right)$$

- Variance of an ensemble of infinitely many predictors with different optimization randomness

- Run multiple seeds on a fixed training set.

Term 2: Variance due to Optimization

- (Mean) Variance due to optimization:

$$\mathbb{E}_S \left[\mathrm{Var}_O \left[h_\theta(x) \mid S \right] \right]$$

- Average (over training sets) variance over optimization randomness for a fixed training set

- Loop on different datasets and average their variances

Term 2: Variance due to Optimization

- (Mean) Variance due to optimization:

$$\mathbb{E}_S \left[\mathrm{Var}_O \left[h_\theta(x) \mid S \right] \right]$$

- Average (over training sets) variance over optimization randomness for a fixed training set

- Loop on different datasets and average their variances

Term 2: Variance due to Optimization

- (Mean) Variance due to optimization:

$$\mathbb{E}_S \left[\mathrm{Var}_O \left[h_\theta(x) \mid S \right] \right]$$

- Average (over training sets) variance over optimization randomness for a fixed training set

- Loop on different datasets and average their variances

Disentangling Variance

- Use the Law of Total Variance:

$$\mathrm{Var}(h_\theta(x)) = \mathrm{Var}_S\left(\mathbb{E}_O[h_\theta(x) \mid S]\right) + \mathbb{E}_S\left[\mathrm{Var}_O(h_\theta(x) \mid S)\right]$$

- Gives a refined 3-term error decomposition:

$$\mathcal{R}_m = \mathcal{E}_{\text{noise}} + \mathcal{E}_{\text{bias}} + \mathcal{E}_{\text{var}}^{\text{sampling}} + \mathcal{E}_{\text{var}}^{\text{optimization}}$$

Disentangling Variance

- Use the Law of Total Variance:

$$\mathrm{Var}(h_\theta(x)) = \mathrm{Var}_S\left(\mathbb{E}_O[h_\theta(x) \mid S]\right) + \mathbb{E}_S\left[\mathrm{Var}_O(h_\theta(x) \mid S)\right]$$

- Gives a refined 3-term error decomposition:

$$\mathcal{R}_m = \mathcal{E}_{\text{noise}} + \mathcal{E}_{\text{bias}} + \mathcal{E}_{\text{var}}^{\text{sampling}} + \mathcal{E}_{\text{var}}^{\text{optimization}}$$

3. Experiments

Experimental Setup: Nested Sampling Scheme

- Create 50 bootstrap replicates S' of dataset S
- Train 100 neural networks for each size (10 seeds and 10 datasets)
- Run for long after 100% accuracy
  - 500 epochs for MNIST
  - 10000 epochs for small MNIST

Experimental Setup: Nested Sampling Scheme

- Create 50 bootstrap replicates S' of dataset S
- Train 100 neural networks for each size (10 seeds and 10 datasets)
- Run for long after 100% accuracy
  - 500 epochs for MNIST
  - 10000 epochs for small MNIST

Experimental Setup: Nested Sampling Scheme

- Create 50 bootstrap replicates S' of dataset S
- Train 100 neural networks for each size (10 seeds and 10 datasets)
- Run for long after 100% accuracy
  - 500 epochs for MNIST
  - 10000 epochs for small MNIST

Experimental Setup: Nested Sampling Scheme

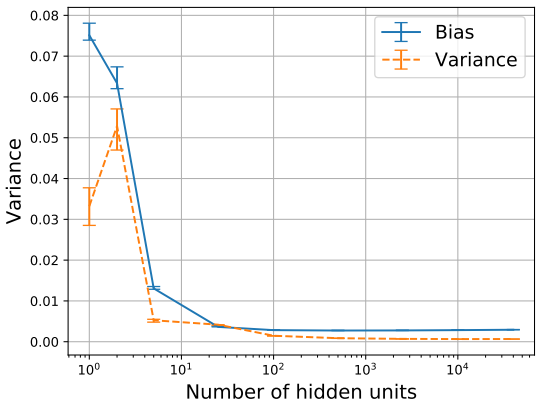
- Create 50 bootstrap replicates S' of dataset S
- Train 100 neural networks for each size (10 seeds and 10 datasets)
- Run for long after 100% accuracy
  - 500 epochs for MNIST
  - 10000 epochs for small MNIST

Experimental Setup: Nested Sampling Scheme

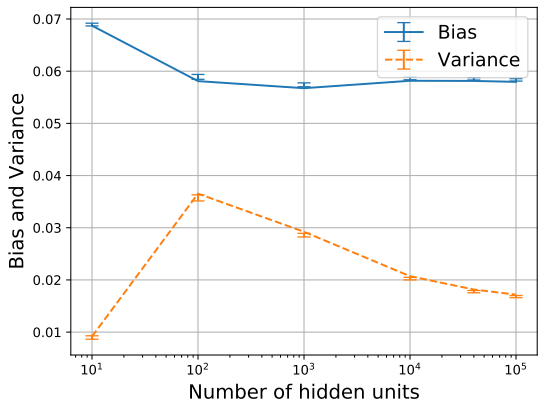
- Create 50 bootstrap replicates S' of dataset S
- Train 100 neural networks for each size (10 seeds and 10 datasets)
- Run for long after 100% accuracy
  - 500 epochs for MNIST
  - 10000 epochs for small MNIST

Decreasing Variance with Increasing Width

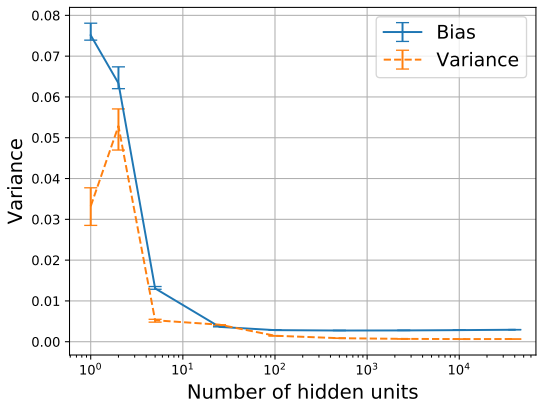
- Variance has unimodal behavior for all datasets



MNIST



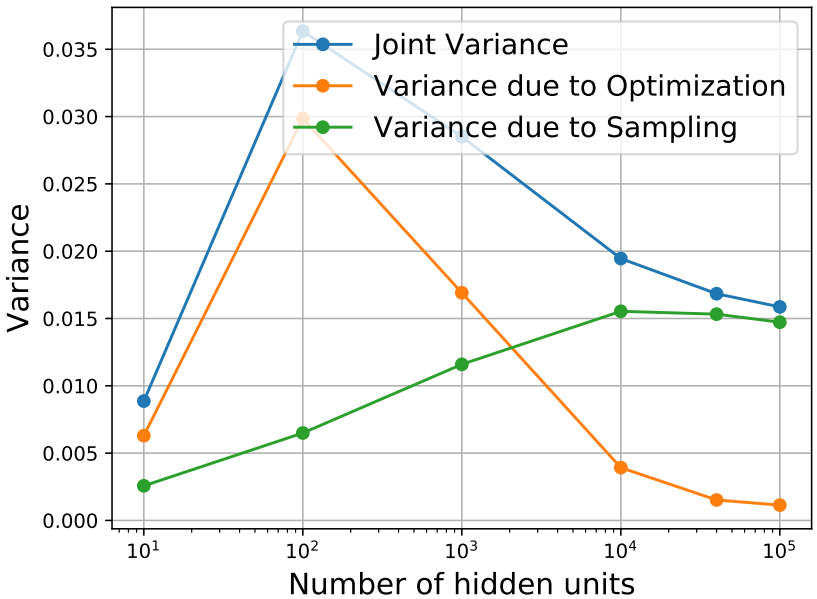
CIFAR10



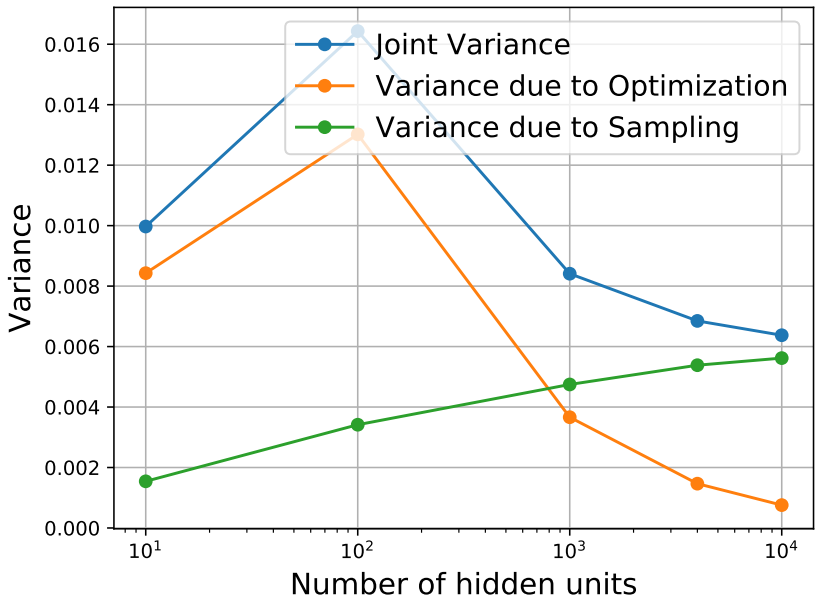
SVHN

Decoupling Variance

- Variance due to optimization decreases with network width
- Variance due to sampling increases slowly and levels off



CIFAR10



SVHN

4. Theoretical Justification: Linear Model

Linear Model Setup

- Least-square linear regression with N parameters and m samples:

$$y = \theta^\top x + \epsilon,$$
$$h_\theta(x) = \hat{\theta}^\top x,$$
$$\mathbb{E}[\epsilon] = 0, \quad \mathsf{Var}(\epsilon) = \sigma_\epsilon^2$$

- Design matrix $X \in \mathbb{R}^{m \times N}$
- Empirical covariance matrix is $\Sigma = X^\top X$
- Quantity of interest: $\mathbb{E}_x \mathsf{Var}_\epsilon(h(x))$

Linear Model Setup

- Least-square linear regression with N parameters and m samples:

$$y = \theta^\top x + \epsilon,$$
$$h_\theta(x) = \hat{\theta}^\top x,$$
$$\mathbb{E}[\epsilon] = 0, \quad \mathsf{Var}(\epsilon) = \sigma_\epsilon^2$$

- Design matrix $X \in \mathbb{R}^{m \times N}$
- Empirical covariance matrix is $\Sigma = X^\top X$
- Quantity of interest: $\mathbb{E}_x \mathsf{Var}_\epsilon(h(x))$

Linear Model Setup

- Least-square linear regression with N parameters and m samples:

$$y = \theta^\top x + \epsilon,$$
$$h_\theta(x) = \hat{\theta}^\top x,$$
$$\mathbb{E}[\epsilon] = 0, \quad \mathsf{Var}(\epsilon) = \sigma_\epsilon^2$$

- Design matrix $X \in \mathbb{R}^{m \times N}$
- Empirical covariance matrix is $\Sigma = X^\top X$
- Quantity of interest: $\mathbb{E}_x \mathsf{Var}_\epsilon(h(x))$

Linear Model Setup

- Least-square linear regression with N parameters and m samples:

$$y = \theta^\top x + \epsilon,$$
$$h_\theta(x) = \hat{\theta}^\top x,$$
$$\mathbb{E}[\epsilon] = 0, \quad \mathsf{Var}(\epsilon) = \sigma_\epsilon^2$$

- Design matrix $X \in \mathbb{R}^{m \times N}$
- Empirical covariance matrix is $\Sigma = X^\top X$
- Quantity of interest: $\mathbb{E}_x \mathsf{Var}_\epsilon(h(x))$

Under-Parameterized Regime ($N \leq m$)

- If X has full rank, Σ is invertible
- The solution $\hat{\theta} = \Sigma^{-1} X^\top Y$
- Variance over all randomness (only ϵ):

$$\mathrm{Var}(h(x)) = \sigma_\epsilon^2 \,\mathrm{Tr}\left(x x^\top \Sigma^{-1}\right)$$

- Taking expected over $x \sim \hat{p}$ gives:

$$\mathbb{E}_{x \sim \hat{p}}[\mathrm{Var}(h(x))] = \frac{N}{m} \sigma_\epsilon^2$$

Under-Parameterized Regime ($N \leq m$)

- If X has full rank, Σ is invertible
- The solution $\hat{\theta} = \Sigma^{-1} X^\top Y$
- Variance over all randomness (only ϵ):

$$\mathrm{Var}(h(x)) = \sigma_\epsilon^2 \,\mathrm{Tr}\left(x x^\top \Sigma^{-1}\right)$$

- Taking expected over $x \sim \hat{p}$ gives:

$$\mathbb{E}_{x\sim\hat{p}}[\mathrm{Var}(h(x))] = \frac{N}{m}\sigma_\epsilon^2$$

Under-Parameterized Regime ($N \leq m$)

- If X has full rank, Σ is invertible
- The solution $\hat{\theta} = \Sigma^{-1} X^\top Y$
- Variance over all randomness (only ϵ):

$$\mathrm{Var}(h(x)) = \sigma_\epsilon^2 \,\mathsf{Tr}\left(xx^\top \Sigma^{-1}\right)$$

- Taking expected over $x \sim \hat{p}$ gives:

$$\mathbb{E}_{x\sim\hat{p}}[\mathrm{Var}(h(x))] = \frac{N}{m}\sigma_\epsilon^2$$

Under-Parameterized Regime ($N \leq m$)

- If X has full rank, Σ is invertible
- The solution $\hat{\theta} = \Sigma^{-1} X^\top Y$
- Variance over all randomness (only ϵ):

$$\mathrm{Var}(h(x)) = \sigma_\epsilon^2 \,\mathsf{Tr}\left(xx^\top \Sigma^{-1}\right)$$

- Taking expected over $x \sim \hat{p}$ gives:

$$\mathbb{E}_{x\sim\hat{p}}[\mathrm{Var}(h(x))] = \frac{N}{m}\sigma_\epsilon^2$$

Over-Parameterized Regime ($N > m$)

- Even for full-rank X, Σ is not invertible
- Gradient descent answer always belong to r-dimensional row(X)
- No learning occurs on $(N - r)$-dimensional null(X)
- The solution $\hat{\theta} = P_{\perp}(\theta_0) + \Sigma^{\dagger} X^{\top} Y$

Over-Parameterized Regime ($N > m$)

- Even for full-rank X, Σ is not invertible
- Gradient descent answer always belong to r-dimensional row(X)
- No learning occurs on $(N - r)$-dimensional null(X)
- The solution $\hat{\theta} = P_{\perp}(\theta_0) + \Sigma^{\dagger} X^{\top} Y$

Over-Parameterized Regime ($N > m$)

- Even for full-rank X, Σ is not invertible
- Gradient descent answer always belong to r-dimensional row(X)
- No learning occurs on $(N - r)$-dimensional null(X)
- The solution $\hat{\theta} = P_{\perp}(\theta_0) + \Sigma^{\dagger} X^{\top} Y$

Over-Parameterized Regime ($N > m$)

- Even for full-rank X, Σ is not invertible
- Gradient descent answer always belong to r-dimensional row(X)
- No learning occurs on $(N - r)$-dimensional null(X)
- The solution $\hat{\theta} = P_{\perp}(\theta_0) + \Sigma^{\dagger} X^{\top} Y$

Over-Parameterized Regime ($N > m$) (contd)

- Variance decomposes into sampling and optimization:

$$\mathsf{Var}(h(x)) = \frac{1}{N}\|P_{\perp}(\theta_0)\|^2 + \sigma_\epsilon^2 \operatorname{Tr}(xx^T\Sigma^\dagger)$$

- Taking expected over $x \sim \hat{p}$ gives:

$$\mathbb{E}_{x\sim\hat{p}}[\mathrm{Var}(h(x))] = \frac{r}{m}\sigma_\epsilon^2$$

Over-Parameterized Regime ($N > m$) (contd)

- Variance decomposes into sampling and optimization:

$$\text{Var}(h(x)) = \frac{1}{N}\|P_{\perp}(\theta_0)\|^2 + \sigma_\epsilon^2 \,\text{Tr}(xx^T\Sigma^\dagger)$$

- Taking expected over $x \sim \hat{p}$ gives:

$$\mathbb{E}_{x\sim\hat{p}}[\text{Var}(h(x))] = \frac{r}{m}\sigma_\epsilon^2$$

5. Theoretical Justification: General Models

General Model Setup & Representation Dimension

- Decompose ambient space $\mathbb{R}^N$ into two orthogonal complement sets
  - Task-related space $\mathcal{M} \subset \mathbb{R}^N$ of dimension $d(N)$
  - Task-irrelevant space $\mathcal{M}^\perp$ of dimension $N - d(N)$
- Each possible parameter $\theta = \theta_{\mathcal{M}} + \theta_{\mathcal{M}^\perp}$

General Model Setup & Representation Dimension

- Decompose ambient space $\mathbb{R}^N$ into two orthogonal complement sets
  - Task-related space $\mathcal{M} \subset \mathbb{R}^N$ of dimension $d(N)$
  - Task-irrelevant space $\mathcal{M}^\perp$ of dimension $N - d(N)$
- Each possible parameter $\theta = \theta_{\mathcal{M}} + \theta_{\mathcal{M}^\perp}$

General Model Setup & Representation Dimension

- Decompose ambient space $\mathbb{R}^N$ into two orthogonal complement sets
  - Task-related space $\mathcal{M} \subset \mathbb{R}^N$ of dimension $d(N)$
  - Task-irrelevant space $\mathcal{M}^\perp$ of dimension $N - d(N)$

- Each possible parameter $\theta = \theta_{\mathcal{M}} + \theta_{\mathcal{M}^\perp}$

General Model Setup & Representation Dimension

- Decompose ambient space $\mathbb{R}^N$ into two orthogonal complement sets
  - Task-related space $\mathcal{M} \subset \mathbb{R}^N$ of dimension $d(N)$
  - Task-irrelevant space $\mathcal{M}^\perp$ of dimension $N - d(N)$

- Each possible parameter $\theta = \theta_{\mathcal{M}} + \theta_{\mathcal{M}^\perp}$

Gaussian Concentration of Measure

Lemma

Let $h : \mathbb{R}^N \to \mathbb{R}$ be a function on the Euclidean space $\mathbb{R}^N$ with Lipschitz constant L; and $\theta \sim \mathcal{N}(0, \sigma I_N)$ sampled from an isotropic N-dimensional Gaussian distribution. Then:

$$\mathbb{P}\left(|h(\theta) - \mathbb{E}[h(\theta)]| \geq \epsilon\right) \leq 2\exp\left(-\frac{C\,\epsilon^2}{L^2\sigma^2}\right)$$

Assumptions

- Assumption 1: The optimization of the loss function is invariant with respect to $\theta_{\mathcal{M}^\perp}$
- Assumption 2: Regardless of initialization, the optimization method consistently yields a solution with the same $\theta_{\mathcal{M}}$ component (i.e. the same vector when projected onto $\mathcal{M}$)

Assumptions

- Assumption 1: The optimization of the loss function is invariant with respect to $\theta_{\mathcal{M}^\perp}$
- Assumption 2: Regardless of initialization, the optimization method consistently yields a solution with the same $\theta_{\mathcal{M}}$ component (i.e. the same vector when projected onto $\mathcal{M}$)

Decay of Initialization Variance due to Initialization

Theorem

In previous setting, let θ denote the parameters at the end of the learning process. Then, for a fixed data set and parameters initialized as $\theta_0 \sim \mathcal{N}(0, \frac{1}{N} I)$, the variance of the prediction satisfies the inequality,

$$Var_{\theta_0}(h_\theta(x)) \leq C \frac{2L^2}{N}$$

where L is the Lipschitz constant of the prediction with respect to θ, and for some universal constant $C > 0$.

- Assume $L = o(\sqrt{N})$, then variance decays as $N \to \infty$.

Variance due to Sampling

- For fixed θ_0, only source of randomness is $\theta^*_{\mathcal{M}}$
- Variance depends only on $\dim(\mathcal{M}) = d(N)$
- For non-increasing $d(N)$ variance does not increase

Variance due to Sampling

- For fixed θ_0, only source of randomness is $\theta^*_{\mathcal{M}}$
- Variance depends only on $\dim(\mathcal{M}) = d(N)$
- For non-increasing $d(N)$ variance does not increase

Variance due to Sampling

- For fixed θ_0, only source of randomness is $\theta^*_{\mathcal{M}}$
- Variance depends only on $\dim(\mathcal{M}) = d(N)$
- For non-increasing $d(N)$ variance does not increase

6. Future Directions & Conclusion

Future Directions

- A probabilistic notion of effective capacity
- How bias and variance change over the course of training

Future Directions

- A probabilistic notion of effective capacity
- How bias and variance change over the course of training

Summary of Core Contributions

- Variance decomposes into two sources
- Variance due to optimization decreases with network width
- Variance due to sampling increases slowly and levels off
- In overparameterized linear model, variance does not depend on N
- In overparameterized general models, under strong assumptions variance is upper bounded

Summary of Core Contributions

- Variance decomposes into two sources
- Variance due to optimization decreases with network width
- Variance due to sampling increases slowly and levels off
- In overparameterized linear model, variance does not depend on N
- In overparameterized general models, under strong assumptions variance is upper bounded

Summary of Core Contributions

- Variance decomposes into two sources
- Variance due to optimization decreases with network width
- Variance due to sampling increases slowly and levels off
- In overparameterized linear model, variance does not depend on N
- In overparameterized general models, under strong assumptions variance is upper bounded

Summary of Core Contributions

- Variance decomposes into two sources
- Variance due to optimization decreases with network width
- Variance due to sampling increases slowly and levels off
- In overparameterized linear model, variance does not depend on N
- In overparameterized general models, under strong assumptions variance is upper bounded

Summary of Core Contributions

- Variance decomposes into two sources
- Variance due to optimization decreases with network width
- Variance due to sampling increases slowly and levels off
- In overparameterized linear model, variance does not depend on N
- In overparameterized general models, under strong assumptions variance is upper bounded

Beyond the Paper

- Double descent (Belkin et al., 2019)
- Analyzed unimodal shape of variance (Yang et al., 2020)
- Decomposed variance into multiple components (D'Ascoli et al., 2020)
- Focus on feature learning dynamics (Pezeshki et al., 2022)
- Connected representation geometry to double descent (Gu, Wang, et al., 2024)

Beyond the Paper

- Double descent (Belkin et al., 2019)
- Analyzed unimodal shape of variance (Yang et al., 2020)
- Decomposed variance into multiple components (D'Ascoli et al., 2020)
- Focus on feature learning dynamics (Pezeshki et al., 2022)
- Connected representation geometry to double descent (Gu, Wang, et al., 2024)

Beyond the Paper

- Double descent (Belkin et al., 2019)
- Analyzed unimodal shape of variance (Yang et al., 2020)
- Decomposed variance into multiple components (D'Ascoli et al., 2020)
- Focus on feature learning dynamics (Pezeshki et al., 2022)
- Connected representation geometry to double descent (Gu, Wang, et al., 2024)

Beyond the Paper

- Double descent (Belkin et al., 2019)
- Analyzed unimodal shape of variance (Yang et al., 2020)
- Decomposed variance into multiple components (D'Ascoli et al., 2020)
- Focus on feature learning dynamics (Pezeshki et al., 2022)
- Connected representation geometry to double descent (Gu, Wang, et al., 2024)

Beyond the Paper

- Double descent (Belkin et al., 2019)
- Analyzed unimodal shape of variance (Yang et al., 2020)
- Decomposed variance into multiple components (D'Ascoli et al., 2020)
- Focus on feature learning dynamics (Pezeshki et al., 2022)
- Connected representation geometry to double descent (Gu, Wang, et al., 2024)

References I

Belkin, M., Hsu, D., Ma, S., & Mandal, S. (2019). Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, *116*(32), 15849–15854. https://doi.org/10.1073/pnas.1903070116

D'Ascoli, S., Refinetti, M., Biroli, G., & Krzakala, F. (2020). Double trouble in double descent: Bias and variance(s) in the lazy regime. *Proceedings of the 37th International Conference on Machine Learning (ICML)*, *119*, 2280–2290.

Geman, S., Bienenstock, E., & Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural computation*, *4*(1), 1–58.

Gu, Y., Wang, Z., et al. (2024). Unraveling the enigma of double descent through the learned feature space. *The Twelfth International Conference on Learning Representations (ICLR)*.

Liang, T., Poggio, T., Rakhlin, A., & Stokes, J. (2019). Fisher-rao metric, geometry, and complexity of neural networks. https://arxiv.org/abs/1711.01530

Neyshabur, B., Tomioka, R., & Srebro, N. (2015). In search of the real inductive bias: On the role of implicit regularization in deep learning. https://arxiv.org/abs/1412.6614

Pezeshki, M., Kaba, S. S., Bengio, Y., Courville, A., Lajoie, G., Sharma, A., et al. (2022). Multi-scale feature learning dynamics: Insights for double descent. *Proceedings of the 39th International Conference on Machine Learning (ICML)*, *162*, 17674–17691.

Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S., & Srebro, N. (2024). The implicit bias of gradient descent on separable data. https://arxiv.org/abs/1710.10345

References II

Yang, Z., Yu, Y., You, C., Steinhardt, J., & Ma, Y. (2020). Rethinking bias-variance trade-off for generalization of neural networks. *Proceedings of the 37th International Conference on Machine Learning (ICML)*, *119*, 10767–10777.

Thank you for your time.

Questions and discussion