

Université de Montréal

**On the Bias-Variance Tradeoff :
Textbooks Need an Update**

by

Brady Neal

Department of Computer Science and Operations Research
Faculty of Arts and Sciences

Thesis submitted for the degree of
Master of Science (M.Sc.)
in Computer Science

10 December 2019

Université de Montréal
Faculty of Graduate and Postdoctoral Studies

This thesis titled

On the Bias-Variance Tradeoff : Textbooks Need an Update

presented by

Brady Neal

was evaluated by a committee composed of the following people:

Aaron Courville
(Committee President)

Ioannis Mitliagkas
(Research Advisor)

Gilles Brassard
(Committee Member)

Abstract

The main goal of this thesis is to point out that the bias-variance tradeoff is not always true (e.g. in neural networks). We advocate for this lack of universality to be acknowledged in textbooks and taught in introductory courses that cover the tradeoff.

We first review the history of the bias-variance tradeoff, its prevalence in textbooks, and some of the main claims made about the bias-variance tradeoff. Through extensive experiments and analysis, we show a lack of a bias-variance tradeoff in neural networks when increasing network width. Our findings seem to contradict the claims of the landmark work by Geman et al. (1992). Motivated by this contradiction, we revisit the experimental measurements in Geman et al. (1992). We discuss that there was never strong evidence for a tradeoff in neural networks when varying the number of parameters. We observe a similar phenomenon beyond supervised learning, with a set of deep reinforcement learning experiments.

We argue that textbook and lecture revisions are in order to convey this nuanced modern understanding of the bias-variance tradeoff.

Keywords: bias-variance tradeoff, neural networks, over-parameterization, generalization

Résumé

L'objectif principal de cette thèse est de souligner que le compromis biais-variance n'est pas toujours vrai (p. ex. dans les réseaux neuronaux). Nous plaidons pour que ce manque d'universalité soit reconnu dans les manuels scolaires et enseigné dans les cours d'introduction qui couvrent le compromis.

Nous passons d'abord en revue l'historique du compromis entre les biais et les variances, sa prévalence dans les manuels scolaires et certaines des principales affirmations faites au sujet du compromis entre les biais et les variances. Au moyen d'expériences et d'analyses approfondies, nous montrons qu'il n'y a pas de compromis entre la variance et le biais dans les réseaux de neurones lorsque la largeur du réseau augmente. Nos conclusions semblent contredire les affirmations de l'œuvre historique de Geman et al. (1992). Motivés par cette contradiction, nous revisitons les mesures expérimentales dans Geman et al. (1992). Nous discutons du fait qu'il n'y a jamais eu de preuves solides d'un compromis dans les réseaux neuronaux lorsque le nombre de paramètres variait. Nous observons un phénomène similaire au-delà de l'apprentissage supervisé, avec un ensemble d'expériences d'apprentissage de renforcement profond.

Nous soutenons que les révisions des manuels et des cours magistraux ont pour but de transmettre cette compréhension moderne nuancée de l'arbitrage entre les biais et les variances.

Mots clés : compromis biais-variance, réseaux de neurones, sur-paramétrage, généralisation

Contents

Abstract	iii
Résumé	iv
List of Figures	ix
List of Acronyms and Abbreviations	xi
Acknowledgements	xii
Chapter 1. Introduction	1
1.1. Motivation.....	1
1.2. Objective of this Thesis	2
1.3. Novel Contributions.....	2
1.4. Related Work	3
1.5. Organization.....	3
Chapter 2. Machine Learning Background	5
2.1. Setting and Notation.....	5
2.2. Generalization	5
2.3. Model Complexity	6
Chapter 3. The Bias-Variance Tradeoff	8
3.1. What are Bias and Variance?	8
3.2. Intuition for the Tradeoff	8
3.3. The Bias-Variance Decomposition	9
3.4. Why do we believe the Bias-Variance Tradeoff?	10
3.4.1. A History	10

3.4.1.1.	Experimental Evidence	10
3.4.1.2.	Supporting Theory.....	12
3.4.2.	The Textbooks.....	12
3.5.	Comparison to the Approximation-Estimation Tradeoff.....	13
3.5.1.	Universal Approximation Theorem for Neural Networks	13
Chapter 4.	The Lack of a Tradeoff	14
4.1.	A Refutation of Geman et al.'s Claims	14
4.2.	Similar Observations in Reinforcement Learning.....	15
4.3.	Previous Work on Boosting	16
4.4.	The Double Descent Curve.....	17
4.5.	The Need to Qualify Claims about the Bias-Variance Tradeoff when Teaching	19
Chapter 5.	A Modern Take on the Bias-Variance Tradeoff in Neural Networks.....	21
5.1.	Introduction	22
5.2.	Related work	24
5.3.	Preliminaries.....	25
5.3.1.	Set-up	25
5.3.2.	Bias-variance decomposition	26
5.3.3.	Further decomposing variance into its sources	27
5.4.	Experiments	28
5.4.1.	Common experimental details.....	28
5.4.2.	Decreasing variance in full data setting.....	29
5.4.3.	Testing the limits: decreasing variance in the small data setting	29
5.4.4.	Decoupling variance due to sampling from variance due to optimization ...	30
5.4.5.	Visualization with regression on sinusoid	30
5.5.	Discussion and theoretical insights	31
5.5.1.	Insights from linear models	31
5.5.1.1.	Under-parameterized setting	31
5.5.1.2.	Over-parameterized setting	32
5.5.2.	A more general result	32

5.5.2.1. Variance due to initialization.....	33
5.5.2.2. Variance due to sampling	33
5.6. Conclusion and future work	34
Acknowledgments	34
Chapter 6. Conclusion and Discussion.....	35
Bibliography	36
Appendix A. Probabilistic notion of effective capacity	42
Appendix B. Additional empirical results and discussion	44
B.1. CIFAR10.....	44
B.2. SVHN.....	45
B.3. MNIST.....	46
B.4. Tuned learning rates for SGD.....	47
B.5. Fixed learning rate results for small data MNIST	47
B.6. Other optimizers for width experiment on small data MNIST	48
B.7. Sinusoid regression experiments.....	49
Appendix C. Depth and variance.....	53
C.1. Main graphs.....	53
C.2. Discussion on need for careful experimental design.....	53
C.3. Vanilla fully connected depth experiments	54
C.4. Skip connections depth experiments	55
C.5. Dynamical isometry depth experiments	55
Appendix D. Some Proofs	56
D.1. Proof of Classic Result for Variance of Linear Model.....	56
D.2. Proof of Result for Variance of Over-parameterized Linear Models	57
D.3. Proof of Theorem 5.5.2.....	58
D.4. Bound on classification error in terms of regression error.....	60

Appendix E. Common intuitions from impactful works	62
--	----

List of Figures

1.1	Mismatch between test error predicted by bias-variance tradeoff and reality	2
2.1	Increasingly complex models fit to sinusoidal data.....	6
3.1	Bias-variance in simple vs. complex hypothesis class.....	9
3.2	Illustration of the bias-variance tradeoff.....	9
3.3	Geman et al. (1992)'s bias-variance experiments on handwritten digits	11
4.1	Bias-variance in simple vs. complex hypothesis class.....	15
4.2	Geman et al. (1992)'s neural network experiments.....	16
4.3	Exponential bias-variance tradeoff in boosting.....	17
4.4	Double descent risk curve conjectured to replace traditional U-shaped curve.....	18
5.1	Traditional bias-variance tradeoff vs. decreasing variance in neural networks.....	23
5.2	Variance due to optimization and variance due to sampling	24
5.3	Small data bias-variance experiments	28
5.4	Visualization of low variance functions learned by large networks.....	29
A.1	Illustration of probabilistic notion of effective capacity	43
B.1	CIFAR10: bias, variance, train error, and test error	44
B.2	CIFAR10 with early stopping: bias, variance, train error, and test error	45
B.3	SVHN: bias, variance, train error, and test error	45
B.4	MNIST: Test error, along with bias and variance	46
B.5	Decomposed variance on MNIST.....	46
B.6	Tuned learning rates used for small data MNIST	47
B.7	Variance on small data with a fixed learning rate of 0.01 for all networks.	47
B.8	Bias-variance experiments with full batch gradient descent	48
B.9	Bias-variance experiments with LBFGS	48

B.10	Visualizations of high variance learners.....	49
B.11	Target function and sampled dataset for sinusoid regression task	49
B.12	Visualization of 100 different functions learned by different width neural networks	50
B.13	Visualization of the mean prediction and variance of different width networks ...	51
B.14	Sinusoid regression task: decomposed variance and test error.....	52
C.1	Main bias-variance with increasing network depth experiment	53
C.2	Increasing depth in vanilla fully connected networks.....	54
C.3	Increasing depth in fully connected networks with skip connections.....	55
C.4	Increasing depth in fully connected networks with dynamical isometry.....	55
E.1	Illustration of common intuition for bias-variance tradeoff (Fortmann-Roe, 2012)	63

List of Acronyms and Abbreviations

CIFAR10 dataset from the Canadian Institute For Advanced Research

KNN K-Nearest Neighbors

LBFGS Limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm

MNIST Modified National Institute of Standards and Technology
(dataset)

SGD Stochastic Gradient Descent

SVHN Street View House Numbers (dataset)

VC dimension Vapnik–Chervonenkis dimension

Acknowledgements

I would like to thank my advisor, Ioannis Mitliagkas, for taking me on as his student, despite the apparent risk that came with that. I greatly appreciate how supportive he has been of me. He has been a fantastic advisor. I would like to thank Yoshua Bengio and Ioannis Mitliagkas for supporting my admission to the department. Without them, I would guess the university would not have accepted me until I completed the final year of my Bachelors degree.

There are many other people at Mila who have been fantastic to interact with. I would like to thank all of the students I worked with, discussed with, and hung out with. I would like to thank Céline Bégin for greatly helping me navigate all the process-related items at a francophone university.

I would like to thank my girlfriend, Isabelle, who has had an immensely positive impact on me throughout my degree.

Chapter 1

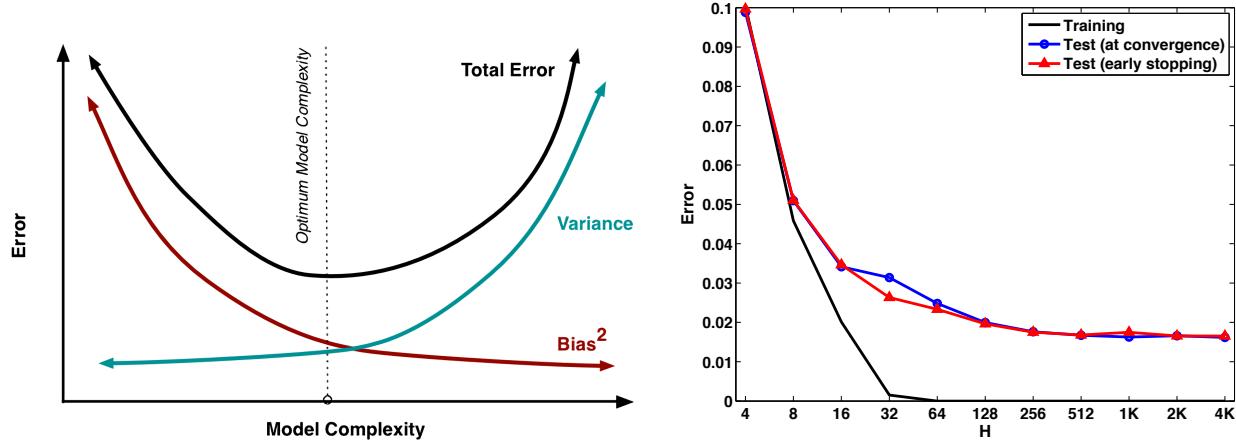
Introduction

1.1. Motivation

An important dogma in machine learning has been that “the price to pay for achieving low bias is high variance” (Geman et al., 1992). This is overwhelmingly the intuition among machine learning practitioners, despite some notable exceptions such as boosting (Schapire & Singer, 1999; Bühlmann & Yu, 2003). The quantities of interest here are the bias and variance of a learned model’s *prediction* on an unseen input, where the randomness comes from the sampling of the training data (see Chapter 3 for more detail). The basic idea is that too simple a model will underfit (high bias) while too complex a model will overfit (high variance) and that bias and variance trade off as model complexity is varied. This is commonly known as the *bias-variance tradeoff* (Figure 1.1(a) and Chapter 3).

A key consequence of the bias-variance tradeoff is that it implies that test error will be a U-shaped curve in model complexity (Figure 1.1(a)). Statistical learning theory (Vapnik, 1998) also predicts a U-shaped test error curve for a number of classic machine learning models by identifying a notion of model capacity, understood as the main parameter controlling this tradeoff. However, there is a growing amount of empirical evidence that *wider* networks generalize *better* than their smaller counterparts (Neyshabur et al., 2015; Zagoruyko & Komodakis, 2016; Novak et al., 2018; Lee et al., 2018; Belkin et al., 2019a; Spigler et al., 2018; Liang et al., 2017; Canziani et al., 2016). In those cases no U-shaped test error curve is observed. In Figure 1.1(b), we depict Neyshabur et al. (2015)’s example of this phenomenon.

The lack of a U-shaped test error curve in these prominent cases suggests that there may be something wrong with the bias-variance tradeoff. In this work, we seek to understand if there really is a bias-variance tradeoff in neural networks when varying the network width by explicitly measuring bias and variance. In their landmark work that highlighted the bias-variance tradeoff in neural networks, Geman et al. (1992) claim that bias decreases and variance increases with network size. This is one of the main claims we refute.



(a) The bias-variance tradeoff predicts a U-shaped test error curve (Fortmann-Roe, 2012).
(b) Neyshabur et al. (2015) found that test error actually decreases with neural network width.

Figure 1.1. Mismatch between test error predicted by bias-variance tradeoff and reality

1.2. Objective of this Thesis

The main objective of this thesis is to show that bias-variance tradeoff thinking can be wrong; researchers and practitioners who assume it to always be true may make incorrect predictions related to model selection. Therefore, we recommend that textbooks and machine learning courses are updated to not present the bias-variance tradeoff as universally true (though, it is accurate for some models, which we review in Section 3.4.1.1). Similarly, we recommend researchers and practitioners update to not universally assume the bias-variance tradeoff (see Section 4.5).

Throughout this thesis, we will reference many textbooks, often using their figures and quoting them. This is to help illustrate what is taught in introductory machine learning courses and to ensure that we are not arguing against strawmen.

1.3. Novel Contributions

- (1) We revisit the bias-variance analysis in the modern setting for neural networks and point out that it is not necessarily a tradeoff as *both* bias *and* variance decrease with network width, yielding better generalization (Section 5.4).
- (2) We perform a more fine-grain study of variance in neural networks by decomposing it into variance due to initialization and variance due to sampling. Variance due to initialization is significant in the under-parameterized regime and monotonically decreases with width in the over-parameterized regime. There, total variance is much lower and dominated by variance due to sampling (Section 5.4.4).

- (3) We remark that this variance phenomenon is already present in over-parameterized linear models. In a simplified setting, inspired by linear models, we provide theoretical analysis in support of our empirical findings ([Section 5.5](#)).

1.4. Related Work

[Neyshabur et al. \(2015\)](#) point out that because increasing network width does not lead to a U-shaped test error curve, there must be some form of implicit regularization controlling capacity. Then, the line of questioning becomes “if not number of parameters, what is the correct measure of model complexity that, when varied, will yield a tradeoff in bias and variance?” [Neyshabur \(2017\)](#); [Neyshabur et al. \(2019\)](#) pursue this direction by studying how test error correlates with different measures of model complexity and by developing models in terms of those new complexity measures.

Our work is consistent with [Neyshabur et al. \(2015\)](#)’s finding, but rather than search for a more appealing measure of model complexity, we study whether it is *necessary* to trade bias for variance. By varying network width (the measure of model complexity that [Geman et al. \(1992\)](#) claimed shows a bias-variance tradeoff), we establish that it is *not necessary* to trade bias for variance when increasing model complexity. To ensure that we are studying networks of increasing capacity, one of the experimental controls we use throughout [Chapter 5](#) is to verify that bias is decreasing.

In concurrent work, [Spigler et al. \(2018\)](#); [Belkin et al. \(2019a\)](#) point out that generalization error acts according to conventional wisdom in the under-parameterized setting, that it decreases with capacity in the over-parameterized setting, and that there is a sharp transition between the two settings. While this transition can roughly be seen as the early hump in variance we observe in some of our graphs, we focus on the over-parameterized setting. [Geiger et al. \(2019a\)](#); [Neyshabur et al. \(2019\)](#); [Liang et al. \(2017\)](#) work toward understanding why increasing over-parameterization does not lead to a U-shaped test error curve. Our work is unique in that we explicitly analyze and experimentally measure the quantities of bias and variance. Interestingly, [Belkin et al. \(2019a\)](#)’s empirical study of test error provides some evidence that our bias-variance finding might not be unique to neural networks and might be found in other models such as decision trees.

1.5. Organization

In [Chapter 2](#), we cover relevant background: the setting in machine learning, the concept of generalization in machine learning, and the concept of model complexity. In [Chapter 3](#), we cover the bias-variance tradeoff in detail, including topics such as why the bias-variance tradeoff is convincing and its relation to the concepts of generalization and model complexity. Then, we argue the bias-variance tradeoff is applied too broadly in [Chapter 4](#) and give specific

recommendations for changes in Section 4.5. In Chapter 5, we provide evidence for this in neural networks.

Chapter 2

Machine Learning Background

2.1. Setting and Notation

We consider the typical supervised learning task of predicting an output $y \in \mathcal{Y}$ from an input $x \in \mathcal{X}$, where the pairs (x, y) are drawn from some unknown joint distribution, \mathcal{D} . The learning problem consists of learning a function $h_S : \mathcal{X} \rightarrow \mathcal{Y}$ from a finite training dataset S of m i.i.d. samples from \mathcal{D} . This learned function is also known as a hypothesis $h \in \mathcal{H}$, which is chosen from a *hypothesis class* \mathcal{H} of possible functions allowed by the model. Then, the learned function h_S and the learning algorithm $\mathcal{A} : (\mathcal{X} \times \mathcal{Y})^m \rightarrow \mathcal{H}$ can be formalized as $h_S \leftarrow \mathcal{A}(S)$. Ideally, we would learn $h_S = f$, where f denotes the “true mapping” from \mathcal{X} to \mathcal{Y} . For some loss function $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$, the quality of a predictor h can be quantified by the *risk* (or *expected error*):

$$R(h) = \mathbb{E}_{(x,y) \sim \mathcal{D}} \ell(h(x), y).$$

The goal in supervised learning is to find $\min_{h \in \mathcal{H}} R(h)$. However, we cannot compute $R(h)$ because we do not know \mathcal{D} . We only have access to the *training error* (a type of *empirical risk*):

$$\hat{R}(h) = \mathbb{E}_{(x,y) \sim S} \ell(h(x), y).$$

This naturally leads to the concept of *empirical risk minimization*: we learn h_S by attempting to minimize $\hat{R}(h_S)$ as a surrogate for $R(h_S)$.

2.2. Generalization

We would like that the learned function h_S generalizes well from the training set $S \sim \mathcal{D}^m$ to other unseen data points drawn from \mathcal{D} . The name “generalization” comes from psychology; for example, if a dog is taught to sit with the verbal cue “sit” by its owner, and then told “sit” by another person, if the dog sits, it would be generalizing. If the dog were to only sit when it hears the exact same sound (made by its owner) it was trained on, it would be “overfitting” and failing to generalize. Overfitting is something to take very seriously in machine learning.

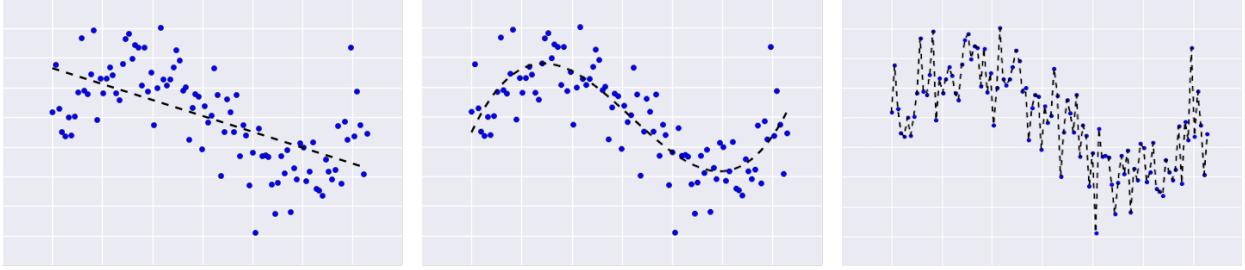


Figure 2.1. Increasingly complex models fit to sinusoidal data (EliteDataScience, 2018)

What can go wrong when minimizing the training error \hat{R} as a surrogate for minimizing the true risk R ? If the hypothesis class \mathcal{H} allows for it, h can fit the data sample too closely, leading to a higher true risk than some h' that has a higher training error than h . More precisely, h can be worse than h' even when $\hat{R}(h) < \hat{R}(h')$ because $R(h) > R(h')$. This can be easily visualized by an example.

In Figure 2.1, we see data coming from a noisy sinusoid task (EliteDataScience, 2018). On the left, a linear model is fit to the data. This leads to both high training error and high true risk. In other words, the model is not complex enough. On the right, a much more complex model is fit to the data. This leads to zero training error, as the learned function fits every training point. However, it will also lead to high true risk as it will not generalize well to unseen data. This is because it is fitting the data too closely, fitting the noise in the data, and, hence, overfitting. The linear model was too simple at the highly complex model was too complex. In the middle, we see a model of about the right complexity that learns a function that will generalize the best of the three.

One notion of generalization that is often seen in statistical learning theory is the *generalization gap*. This is simply the difference between the true risk and the training error:

$$\mathcal{E}_{\text{gap}}(h) = R(h) - \hat{R}(h) \quad (2.2.1)$$

2.3. Model Complexity

In Figure 2.1, the key concept that varies (increases from left to right) is *model complexity* (i.e., the complexity of the hypothesis class \mathcal{H}). Models that are not sufficiently complex will underfit, while models that are too complex will overfit (see, e.g., Figure 2.1). In terms of hypothesis classes, the larger $|\mathcal{H}|$ is, the more functions exist that will fit the training data, but they might not perform well on unseen data. It is intuitive that the larger $|\mathcal{H}|$ is, the more our model will overfit (see Figure 3.1 in Section 3.2 for an illustration of this in the bias-variance framework). And indeed, there is theory that supports this intuition (Mohri

et al., 2012, Theorem 2.2). For any $\delta > 0$, with probability at least $1 - \delta$,

$$\forall h \in \mathcal{H}, \quad R(h) \leq \hat{R}(h) + \sqrt{\frac{\log |\mathcal{H}| + \log \frac{2}{\delta}}{2m}}. \quad (2.3.1)$$

The quantity $|\mathcal{H}|$ is one notion of model complexity. However, for many models (e.g. neural networks), $|\mathcal{H}|$ is infinite. Therefore, a better notion of model complexity is needed. The VC dimension of \mathcal{H} , $VC(\mathcal{H})$, is a better notion of model complexity, which leads to a finite bound with infinite models classes (Mohri et al., 2012, Chapter 3.3). For any $\delta > 0$, with probability at least $1 - \delta$,

$$\forall h \in \mathcal{H}, \quad R(h) \leq \hat{R}(h) + \sqrt{\frac{8VC(\mathcal{H}) \log \frac{2em}{VC(\mathcal{H})} + 8 \log \frac{4}{\delta}}{m}}. \quad (2.3.2)$$

This bounds grows with $VC(\mathcal{H})$. For many models the VC dimension ends up being roughly proportional to the number of parameters in the model. For example, the VC dimension of various kinds of neural networks grows with the number of parameters (Baum & Haussler, 1989; Karpinski & Macintyre, 1995; Bartlett et al., 1998; Harvey et al., 2017).

Rademacher complexity is another measure of model complexity. Intuitively, it measures the capacity of a model to fit random noise. Generalization bounds in terms of Rademacher complexity are also prevalent (Mohri et al., 2012, Theorem 3.2):

$$\forall h \in \mathcal{H}, \quad R(h) \leq \hat{R}(h) + \mathfrak{R}_m(\mathcal{H}) + \sqrt{\frac{\log \frac{1}{\delta}}{2m}} \quad (2.3.3)$$

where $\mathfrak{R}_m(\mathcal{H})$ denotes the Rademacher complexity of \mathcal{H} . Known bounds on Rademacher complexity also grow with the number of parameters (Bartlett & Mendelson, 2003).

These generalization bounds in terms of model complexity are important because of how they are interpreted. The general idea is that the model must be complex enough to achieve a low $\hat{R}(h)$, but not too complex that the complexity measures such as $VC(\mathcal{H})$ and $\mathfrak{R}_m(\mathcal{H})$ will blow up, leading to high bounds on $R(h)$. For example, when interpreting the VC-based generalization bound, Abu-Mostafa et al. (2012, Chapter 2.2) wrote, “Although the bound is loose, it tends to be equally loose for different learning models, and hence is useful for comparing the generalization performance of these models. [...] In real applications, learning models with lower $VC(\mathcal{H})$ tend to generalize better than those with higher $VC(\mathcal{H})$. Because of this observation, the VC analysis proves useful in practice [...] the VC bound can be used as a guideline for generalization, relatively if not absolutely.”

Chapter 3

The Bias-Variance Tradeoff

3.1. What are Bias and Variance?

The bias is a measure of how close the central tendency of a learner is to the true function f . If, on average (over training sets S), the learner learns the true function f , then the learner is unbiased. For some $x \sim \mathcal{D}$, the bias is

$$\text{Bias}(h_S) = \mathbb{E}_S[h_S(x)] - f(x).$$

The variance is a measure of fluctuations of a learner around its central tendency, where the fluctuations result from different samplings of the training set. By definition, a learner that generalizes well does not learn dramatically different functions, depending on sampling of the training set. Let $\mathcal{Y} = \mathbb{R}$ for simplicity; then for some $x \sim \mathcal{D}$, the variance is

$$\text{Var}(h_S) = \mathbb{E}_S \left[(h_S(x) - \mathbb{E}_S[h_S(x)])^2 \right].$$

3.2. Intuition for the Tradeoff

Similar to the idea that larger hypothesis classes lead to overfitting in Section 2.3, in the bias-variance context, there is the idea that larger hypothesis classes lead to higher variance. This is illustrated in Figure 3.1, which comes from Abu-Mostafa et al. (2012). Also, illustrated is the idea that bias decreases when increasing the size of the hypothesis class because there will be more hypotheses that are closer to the true function f . In Figure 3.1, a hypothesis class that contains only a single hypothesis is depicted on the left; this will, of course, lead to bias as that hypothesis does not match f , but it will also lead to zero variance, which is a positive. In contrast, on the right, there is a larger hypothesis class (Figure 3.1); in this example, this leads to nearly zero bias, but it comes at the expense of incurring variance. The bottom of Figure 3.1 is shorthand that summarizes the idea that when you increase the size of the hypothesis class, you decrease bias and increase variance. Figure 2.1 is another example of this: the leftmost learner has high bias and low variance,

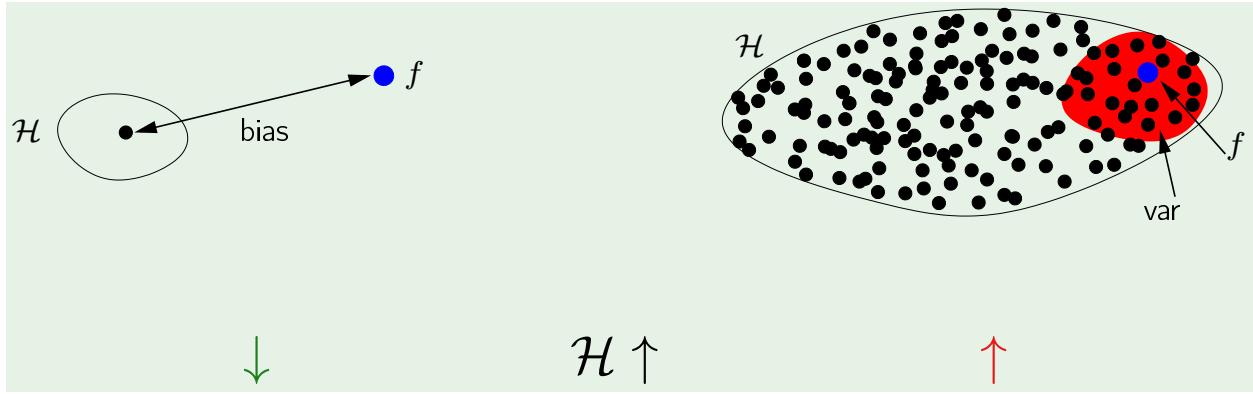


Figure 3.1. Bias-variance in simple vs. complex hypothesis class (Abu-Mostafa et al., 2012)

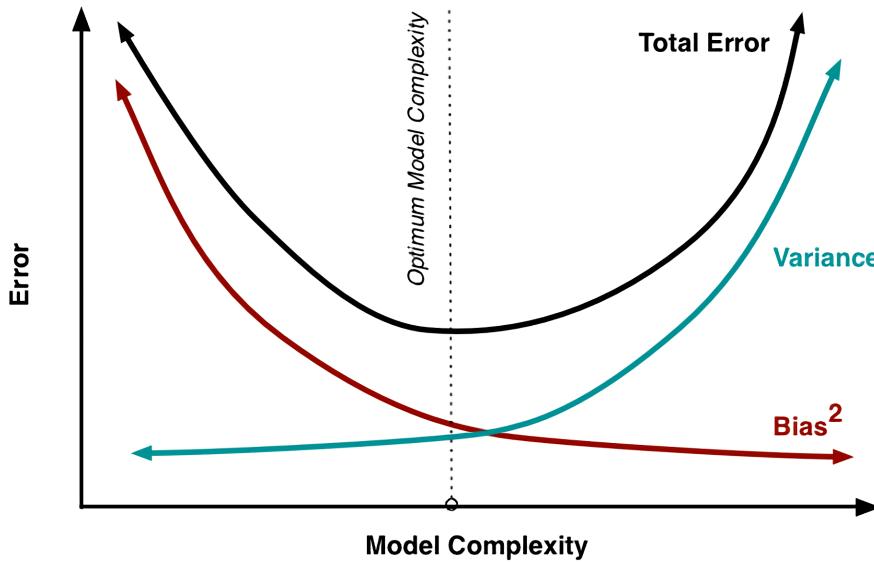


Figure 3.2. Illustration of the bias-variance tradeoff (Fortmann-Roe, 2012)

the rightmost learner has low bias and high variance, and the learner in the middle has something close to the optimal balance of bias and variance.

In their landmark paper, Geman et al. (1992) capture the essence of the bias-variance tradeoff with the following claim: “the price to pay for achieving low bias is high variance.” In Figure 3.2, we see the common illustration of the bias-variance tradeoff (Fortmann-Roe (2012)). Note the important U shape of the test error curve with increasing model complexity. The idea is that the optimal point on that U can be achieved by achieving the optimal balance of bias and variance. This tradeoff hypothesis is ubiquitous, as we will see in Section 3.4.2.

3.3. The Bias-Variance Decomposition

Geman et al. (1992) considered the average case (over training sets) quantity $\mathbb{E}_S R(h_S)$ with squared-loss and showed that it can be cleanly decomposed into bias and variance

components:

$$\mathbb{E}_S R(h_S) = \mathcal{E}_{\text{bias}}(h_S) + \mathcal{E}_{\text{variance}}(h_S) + \mathcal{E}_{\text{noise}} \quad (3.3.1)$$

Although a decomposition does not prove that the bias-variance tradeoff is true, it does show that the average error is made up of a sum of bias and variance components. Then, if the average error is held constant and bias is varied, variance must also vary (and vice versa). This greatly added to the strong intuition of the bias-variance tradeoff, and [Geman et al. \(1992\)](#) became quite highly cited for their contribution.

Note that risks computed with classification losses (e.g cross-entropy or 0-1 loss) do not have such a clean, additive bias-variance decomposition ([Domingos, 2000](#); [James, 2003](#)). However, because the concept of a tradeoff is not reliant on an additive decomposition (see [Hastie & Tibshirani \(1990, Chapter 3.3\)](#) for presence of the bias-variance tradeoff before the bias-variance decomposition), the concept of the bias-variance tradeoff is applied extremely broadly (see [Section 3.4.2](#)), including settings where an additive decomposition does not seem possible.

3.4. Why do we believe the Bias-Variance Tradeoff?

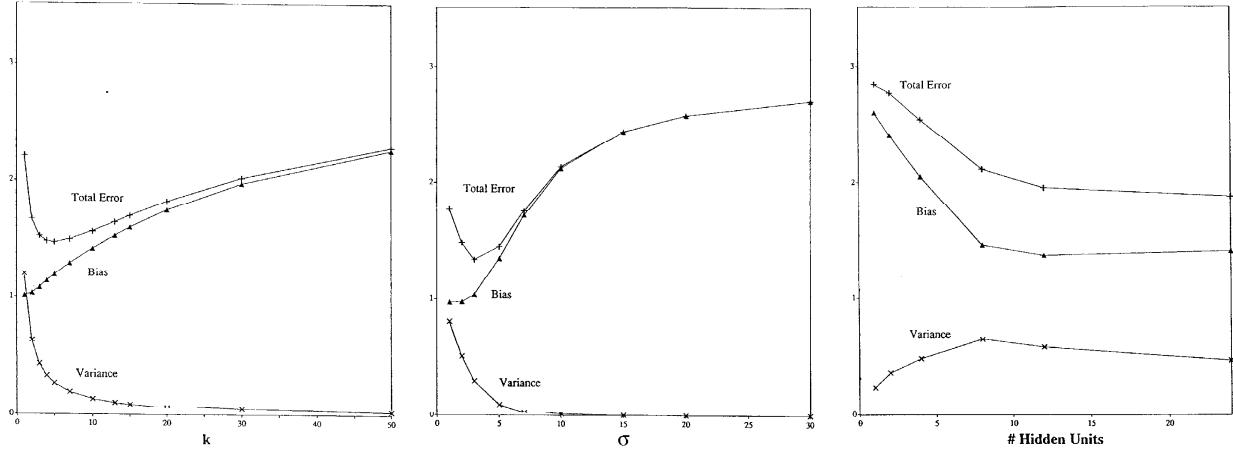
A universal bias-variance tradeoff, without qualifications, is a mere hypothesis. Potentially, its most significant appeal comes from its intuitiveness. In this section, we review the history of the bias-variance tradeoff, the evidence in support of it, and its prevalence in textbooks (which also seem to contain much of the authoritative evidence).

3.4.1. A History

The concept that we know as the “bias-variance tradeoff” in machine learning has a long history, with its basis in statistics. *Neural Networks and the Bias/Variance Dilemma* ([Geman et al., 1992](#)) is the most cited work largely because it introduced the bias-variance decomposition to the machine learning community, provided convincing experiments with nonparametric methods, and popularized the bias-variance tradeoff in the neural network and machine learning community. However, the bias-variance tradeoff was already present in a textbook in 1990 ([Hastie & Tibshirani, 1990](#)), and it dates back at least as far back as 1952 in statistics when [Grenander \(1952\)](#) referred to the concept as an “uncertainty principle.”

3.4.1.1. Experimental Evidence

[Geman et al. \(1992\)](#) ran experiments using two nonparametric methods ([KNN and kernel regression](#)) and neural networks on a partially corrupted version of the handwritten digits [Guyon \(1988\)](#) collected ([Figure 3.3](#)). The experiments on k-nearest neighbor (KNN) ([Figure 3.3\(a\)](#)) and kernel regression ([Figure 3.3\(b\)](#)) yield clear bias-variance tradeoff curves with



(a) K-nearest neighbor (KNN) (b) Kernel regression (higher kernel width σ is less complexity) (c) Single hidden layer neural network (higher “# Hidden Units” is more complexity)

Figure 3.3. Geman et al. (1992)’s bias-variance experiments on handwritten digits

U-shaped test error curves in their respective complexity parameters k and σ . The experiment with neural networks (Figure 3.3(c)) is substantially less conclusive. Geman et al. (1992) maintain their claim that there is a bias-variance tradeoff in neural networks and explain their inconclusive experiments as a result of convergence issues:

«*The basic trend is what we expect: bias falls and variance increases with the number of hidden units. The effects are not perfectly demonstrated (notice, for example, the dip in variance in the experiments with the largest numbers of hidden units), presumably because the phenomenon of overfitting is complicated by convergence issues and perhaps also by our decision to stop the training prematurely.*»

This is the first glimpse we see of the cracks in the bias-variance tradeoff hypothesis.

There is a fair amount of empirical evidence for the bias-variance tradeoff in different complexity parameters in a variety of methods. Wahba & Wold (1975) show a tradeoff in complexity with cubic splines when varying their smoothing parameter. Geurts (2002, Section 5.4.3) show a bias-variance tradeoff in decision trees when varying tree size. Hastie & Tibshirani (1990, Chapter 3.3) show a bias-variance tradeoff with a “running-mean smoother” (i.e. KNN, but in statistics) when varying k . Bishop (2006, Chapter 3.2) show a bias-variance tradeoff with Gaussian basis function linear regression when varying the L2 regularization (weight decay) parameter λ . Goodfellow et al. (2016, Chapters 5.2 and 5.3) show a tradeoff in complexity with fitting polynomials when varying either the degree or the L2 regularization parameter.

3.4.1.2. Supporting Theory

The first main theory that supports the bias-variance tradeoff is the [sample of generalization upper bounds](#) presented in [Section 2.3](#) that grow with the number of parameters. This is simply because the bias-variance tradeoff is a clear conceptualization of one of the common interpretations of those bounds: models that are too simple will not perform well due to underfitting (high bias or high training error \hat{R}) while models that are too complex will not perform well due overfitting (high variance caused by high model complexity such as $VC(\mathcal{H})$). However, it should be noted that these upper bounds do not guarantee that a model with high $VC(\mathcal{H})$ will actually have high variance (or high test error); a lower bound for cases seen in practice (not worst case) would be needed for that.

[Hastie et al. \(2001, Chapter 7.3\)](#) show that you can derive closed-form expressions for the variance for simple models such as KNN and linear regression in the “fixed-design” setting where the design matrix X is fixed. In this setting, $Y = f(X) + \epsilon$, where f is the true function mapping examples x (rows in X) to y (elements in the vector Y). Then, the randomness in Y is determined by the zero-mean random variable ϵ . In this setting, [Hastie et al. \(2001, Chapter 7.3\)](#) show that variance for KNN models scales as $\frac{1}{K}$. Similarly, they show that the variance for linear regression grows linearly with the number of parameters, assuming $X^T X$ is invertible. Note that in the over-parameterized setting ($X^T X$ is not invertible), we show that the variance of linear regression does not grow with the number of parameters (see [Section 5.5.1](#) in [Chapter 5](#)).

Because neural networks are much more complicated models than KNN and linear regression, we must resort to bounds on the variance of a neural network. [Barron \(1994\)](#) derives an upper bound on the estimation error of a single hidden layer neural network that grows linearly with the number of hidden units. The estimation error is not the same thing as variance, but it is analogous (see [Section 3.5](#)). This kind of bound is similar to the bounds described in [Section 2.3](#). Again, it should be noted that because this is an upper bound, it does not actually imply that large neural networks will have high estimation error.

3.4.2. The Textbooks

The concept of the bias-variance tradeoff is ubiquitous, appearing in many of the textbooks that are used in machine learning education: [Hastie et al. \(2001, Chapters 2.9 and 7.3\)](#), [Bishop \(2006, Chapter 3.2\)](#), [Goodfellow et al. \(2016, Chapter 5.4.4\)](#)), [Abu-Mostafa et al. \(2012, Chapter 2.3\)](#), [James et al. \(2014, Chapter 2.2.2\)](#), [Hastie & Tibshirani \(1990, Chapter 3.3\)](#), [Duda et al. \(2001, Chapter 9.3\)](#). Here are two excerpts:

- “As a general rule, as we use more flexible methods, the variance will increase and the bias will decrease. The relative rate of change of these two quantities determines whether the test MSE increases or decreases. As we increase the flexibility of a class

- of methods, the bias tends to initially decrease faster than the variance increases. Consequently, the expected test MSE declines. However, at some point increasing flexibility has little impact on the bias but starts to significantly increase the variance. When this happens the test MSE increases” (James et al., 2014, Chapter 2.2.2).
- “As the model complexity of our procedure is increased, the variance tends to increase and the squared bias tends to decrease” (Hastie et al., 2001, Chapters 2.9).

3.5. Comparison to the Approximation-Estimation Tradeoff

The bias-variance tradeoff is not the only tradeoff in machine learning that is related to generalization. For example, when h_S is chosen from a hypothesis class \mathcal{H} , $R(h_S)$ can be decomposed into approximation error and estimation error:

$$R(h_S) = \mathcal{E}_{\text{app}} + \mathcal{E}_{\text{est}}$$

where $\mathcal{E}_{\text{app}} = \min_{h \in \mathcal{H}} R(h)$ and $\mathcal{E}_{\text{est}} = R(h_S) - \mathcal{E}_{\text{app}}$. Shalev-Shwartz & Ben-David (2014, Section 5.2) present this decomposition and frame it as a tradeoff. Bottou & Bousquet (2008) describe this as the “well known tradeoff between approximation error and estimation error” and present it in a slightly more lucid way as a decomposition of the *excess risk*:

$$\mathbb{E}[R(h_S) - R(h^*)] = \mathbb{E}[R(h_{\mathcal{H}}^*) - R(h^*)] + \mathbb{E}[R(h_S) - R(h_{\mathcal{H}}^*)]$$

where $R(h^*)$ is the Bayes error and $h_{\mathcal{H}}^* = \arg \min_{h \in \mathcal{H}} R(h)$ is the best hypothesis in \mathcal{H} . The approximation error can then be interpreted as the distance of the best hypothesis in \mathcal{H} from the Bayes classifier, and the estimation error can be interpreted as the average distance of the learned hypothesis from the best hypothesis in \mathcal{H} . It is common to associate larger \mathcal{H} with smaller approximation error and larger estimation error, just like it is common to associate larger \mathcal{H} with smaller bias and larger variance. While bias (variance) and approximation error (estimation error) are qualitatively similar, they are not the exact same.

3.5.1. Universal Approximation Theorem for Neural Networks

The commonly cited universal approximation property of neural networks (Cybenko, 1989; Hornik, 1991; Leshno & Schocken, 1993) means that the approximation error goes to 0 as the network width increases; these results do not say anything about estimation error. In other words, the universal approximation error does not imply that wider networks are better. It implies that wider networks yield lower *approximation error*; traditional thinking suggests that this also means that wider networks yield *higher estimation error*.

Chapter 4

The Lack of a Tradeoff

4.1. A Refutation of Geman et al.'s Claims

In their highly influential paper, [Geman et al. \(1992\)](#) make several claims of varying specificity. We will start with their most general claim, which is simply a statement of the bias-variance tradeoff:

«*the price to pay for achieving low bias is high variance.* »

However, this claim is not true in the general case. The fact that the expected risk can be decomposed into squared bias and variance does imply that the two terms trade off. It is not necessary to trade bias for variance in all settings. For [example, it is not necessary to trade bias for variance in neural networks](#) ([Neal et al. \(2018\)](#), Chapter 5).

As neural networks are a main focus of [Geman et al. \(1992\)](#)'s work, they also make a clear claim relevant to neural networks: “bias falls and variance increases with the number of hidden units.” We directly test this claim in [Chapter 5](#) and show it to be false on a variety of datasets. *Both* bias *and* variance can decrease as network width increases. In [Figure 4.1](#) we contrast the common intuition about the bias-variance tradeoff (left, as inspired by [Figure 3.1](#)) with what we observe in neural networks (right). This is a specific example of not having to pay any price of increased variance when decreasing bias.

In fact, [Geman et al. \(1992\)](#)'s own experiments with neural networks do not even support their claim that “bias falls and variance increases with the number of hidden units.” [Geman et al. \(1992, Figures 16 and 8\)](#) run experiments with a handwritten digit recognition dataset ([Figure 4.2\(a\)](#)) and with a sinusoid dataset ([Figure 4.2\(b\)](#)). In both of these datasets, they see decreasing variance when increasing the number of hidden units ([Figure 4.2](#)). [Geman et al. \(1992, page 33\)](#) explain this seeming evidence against their claim as a product of “convergence issues,” and maintain their claim: “The basic trend is what we expect: bias falls and variance increases with the number of hidden units. The effects are not perfectly demonstrated (notice, for example, the dip in variance in the experiments with the largest

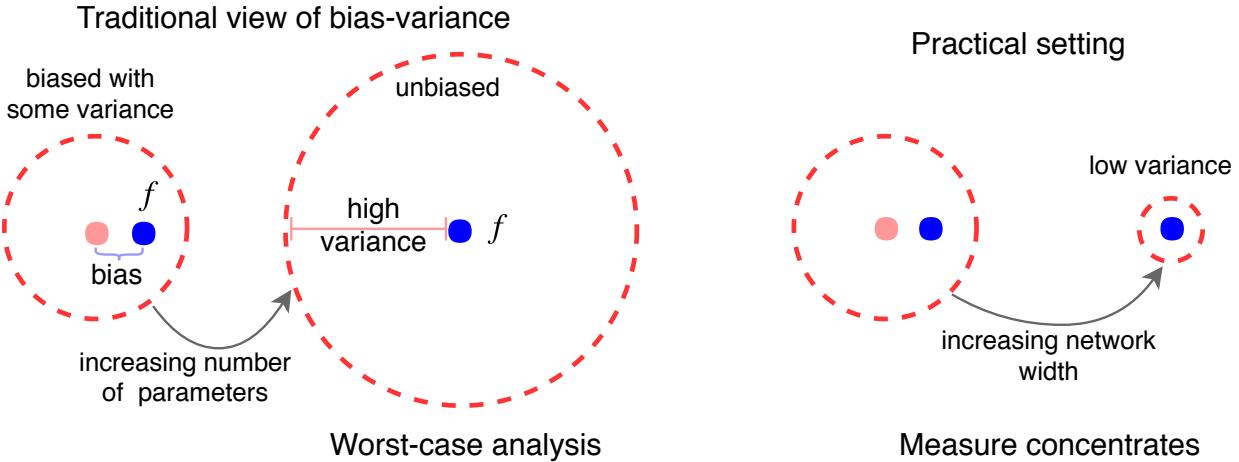


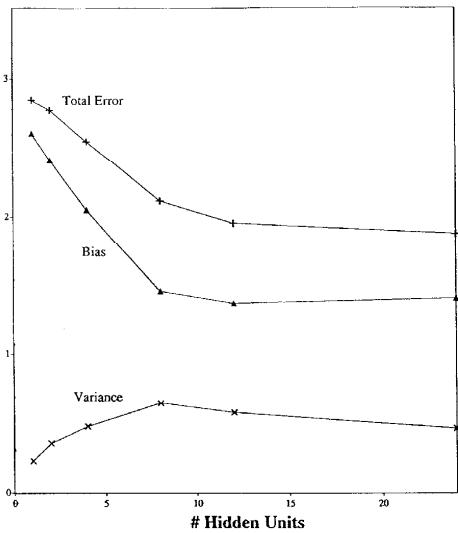
Figure 4.1. Bias-variance in simple vs. complex hypothesis class

numbers of hidden units), presumably because the phenomenon of overfitting is complicated by convergence issues and perhaps also by our decision to stop the training prematurely.”

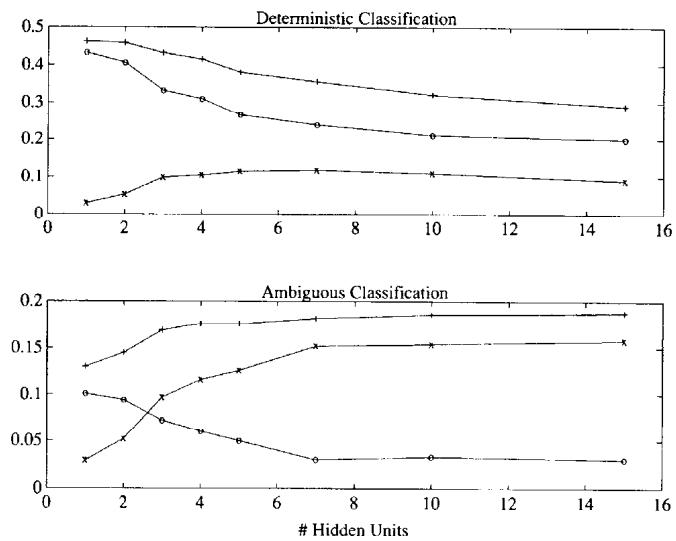
In their paper, Geman et al. (1992) give the following prescription for choosing the width of a neural network: “How big a network should we employ? A small network, with say one hidden unit, is likely to be biased, since the repertoire of available functions spanned by $f(x; w)$ over allowable weights will in this case be quite limited. If the true regression is poorly approximated within this class, there will necessarily be a substantial bias. On the other hand, if we overparameterize, via a large number of hidden units and associated weights, then the bias will be reduced (indeed, with enough weights and hidden units, the network will interpolate the data), but there is then the danger of a significant variance contribution to the mean-squared error.” Although this fits the conventional wisdom laid out in Section 2.3 and Chapter 3, we find this way of thinking to be misleading, leading researchers to incorrect predictions.

4.2. Similar Observations in Reinforcement Learning

Neyshabur et al. (2015) found that increasing the width of a single hidden layer neural network leads to decreasing test error on MNIST and CIFAR10 until it levels off (never going back up). We explore whether this phenomenon extends to deep reinforcement learning. We provide some evidence that it does, finding that wider networks do seem to perform better than their smaller counterparts in deep reinforcement learning as well (Neal & Mitliagkas, 2019). Combining that with the results from Chapter 5, we infer that very wide networks do not have to suffer from high variance (in exchange for low bias) in reinforcement learning either.



(a) Bias, variance, and total error as a function of number of hidden units in Geman et al. (1992)'s handwritten digit recognition experiment. Note that variance is decreasing with width in roughly the last $\frac{2}{3}$ of the graph.



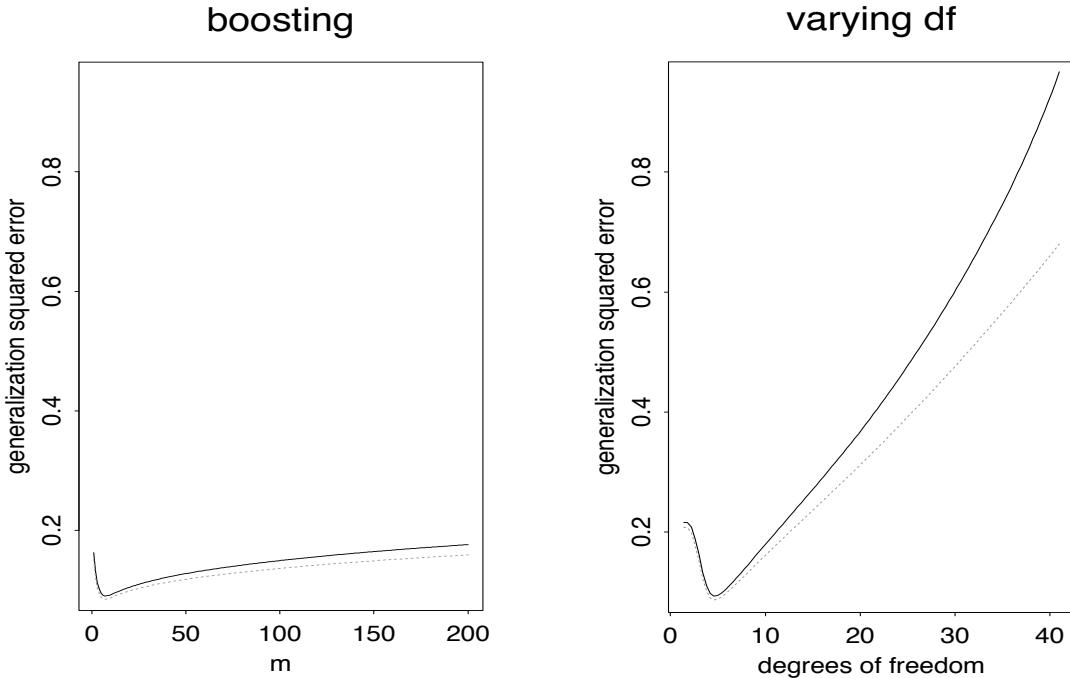
(b) Bias (o), variance (x), and total error (+) as a function of number of hidden units in Geman et al. (1992)'s noise-free (deterministic) and noisy (ambiguous) sinusoid classification experiments. Note that decreasing variance is already seen in the deterministic (top) experiment with neural networks as small as 7-15 hidden units.

Figure 4.2. Geman et al. (1992)'s neural network experiments

4.3. Previous Work on Boosting

Bühlmann & Yu (2003)'s work on the bias-variance tradeoff in boosting is motivated by “boosting’s resistance to overfitting” when increasing the number of iterations. For example, in Schapire & Singer (1999, Figures 8-10), they run experiments on many datasets, finding that, on some datasets, test error decreases and plateaus without increasing with more iterations of boosting. This is only the case in some of their experiments, as in roughly half of their experiments, Schapire & Singer (1999) find that test error does eventually increase with number of iterations of boosting. Still, because roughly half of the experiments show “boosting’s resistance to overfitting,” Bühlmann & Yu (2003) study bias-variance in boosting.

Bühlmann & Yu (2003) find that the lack of increasing test error when increasing number of iterations (“boosting’s resistance to overfitting”) can be explained in terms of bias and variance. In Theorem 1, they show exponentially decaying bias and variance that grows at an exponentially decaying rate with number of iterations. There are some specifics to this that are related to the strength/weakness of the learner that is boosted, but this is how they explain why monotonically decreasing test error can sometimes be seen when increasing the number of iterations in boosting: “(2) Provided that the learner is sufficiently weak, boosting always improves, as we show in Theorem 1” (Bühlmann & Yu, 2003, Section 3.2.2).



(a) Test error as a function of the number of iterations of boosting. Although it decreases and then increases, it increases at a much slower rate than the classic bias-variance tradeoff would suggest (see figure on the right). [Bühlmann & Yu \(2003\)](#) call this slower increase an “a new exponential bias-variance tradeoff.”

(b) Test error as a function of the amount of smoothing in a cubic spline. This is the test error curve that the classic bias-variance tradeoff would suggest; test error increases with the complexity parameter at a rate that is at least linear.

Figure 4.3. Exponential bias-variance tradeoff in boosting ([Bühlmann & Yu, 2003](#))

All this said, [Bühlmann & Yu \(2003\)](#)’s work should not be interpreted as showing a lack of a bias-variance tradeoff in boosting. Rather, it shows a bias-variance tradeoff where the growth of variance with the complexity parameter is exponentially smaller than that in the traditional bias-variance tradeoff (see [Figure 4.3](#)), which implies that variance does not grow forever when increasing the number of iterations. Their work is an important example of a departure from the conventional bias-variance tradeoff.

4.4. The Double Descent Curve

A conjecture that has recently gained popularity is the idea the risk behaves as a “double descent” curve in model complexity ([Figure 4.4](#)). Specifically, the idea is that the risk behaves according to the classical bias-variance tradeoff wisdom ([Chapter 3](#)) in the under-parameterized regime; the risk decreases with model complexity in the over-parameterized

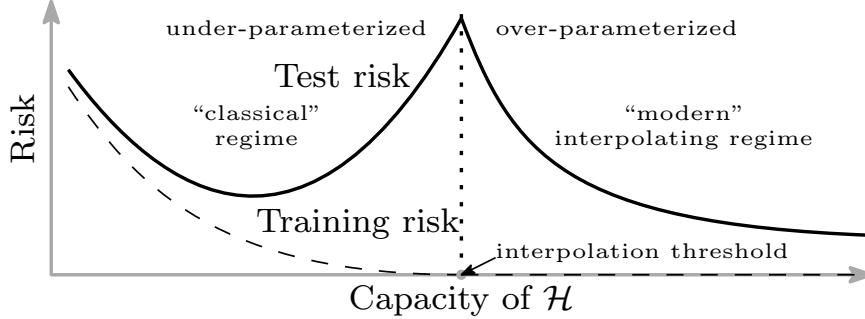


Figure 4.4. Double descent curve, showing U-shaped risk curve in under-parameterized regime and decreasing curve in over-parameterized regime (Belkin et al., 2019a).

regime; and there is a sharp transition from the under-parameterized regime to the over-parameterized regime where the training error is 0. Belkin et al. (2019a) illustrate this in Figure 4.4.

In previous work, Advani & Saxe (2017) observed this phenomenon in linear student-teacher¹ networks and with nonlinear networks on MNIST. In concurrent (to Chapter 5) work, Spigler et al. (2018); Geiger et al. (2019b); Belkin et al. (2019a) also studied this phenomenon. Spigler et al. (2018); Geiger et al. (2019b) described the cusp in the double descent curve as corresponding to a phase transition and draw the analogy to the “jamming transition” in particle systems. Belkin et al. (2019a) conjectured that this phenomenon is fairly general (as opposed to just being restricted to neural networks). Belkin et al. (2019a) showed the phenomenon in random forests, in addition to neural networks, and coined the term “double descent.” Nakkiran et al. (2019) recently showed that this double descent phenomenon is present in many state-of-the-art architectures such as convolutional neural networks, ResNets, and transformers, as opposed to only being present in more toy settings. The double descent phenomenon in simple settings such as shallow linear models can be seen in work that dates as far back as 1995 (Opper, 1995, 2001; Bös & Opper, 1997).

Our work in Chapter 5 is consistent with the double descent curve. Although we were not looking for the cusp in the double descent curve (can require dense sampling of model sizes and specific experimental details), we do seem to see it in several variance figures in Chapter 5. All the works on the double descent curve examine the risk (or test error). In order to test the bias-variance hypothesis, it is important to actually measure bias and variance because test error and bias can decrease while variance still increases at an exponentially decaying rate (Section 4.3).

1. “Teacher” here refers to the fact that the data is generated by a neural network.

4.5. The Need to Qualify Claims about the Bias-Variance Tradeoff when Teaching

Students who take introductory machine learning courses are typically taught the bias-variance tradeoff as a general, unavoidable truth that applies anywhere there is some notion of increasing model complexity (see [Section 3.4.2](#) for its prevalence and representative quotes from textbooks). This leads machine learning experts to sometimes make incorrect inferences about model selection with high confidence. From extensive personal communication, it appears that most researchers unfamiliar with work like [Neyshabur et al. \(2015\)](#)'s react with incredulity to the results described in [Section 4.1](#), [Section 4.2](#), and [Chapter 5](#). Even among researchers who have familiarized themselves with [Neyshabur et al. \(2015\)](#)'s results on test error, one can find many who are surprised by our results. We attribute this phenomenon to the strong influence [Geman et al. \(1992\)](#)'s claims have had on the research community. In other words, a sizable portion of researchers can be dogmatic about the conventional tradeoff wisdom described in [Section 2.3](#) and [Chapter 3](#). Qualifying the conventional tradeoff wisdom in textbooks and introductory courses by noting that the tradeoff intuition is useful sometimes and misleading other times would prevent future students from subscribing to this intuitive dogma.

The goal in amending textbooks/lectures that teach the bias-variance tradeoff is to make it more clear that the bias-variance tradeoff is not a universal truth. Here, we present three simple qualifications that, if integrated, would more accurately represent the evidence we have on the bias-variance tradeoff and would help prevent students from interpreting that the bias-variance tradeoff is universal:

- (1) Expected error can be decomposed into (squared) bias and variance, when using squared loss ([Geman et al., 1992](#)), but this **decomposition does not imply a tradeoff**. This lack of implication should be made explicit in textbooks because the decomposition is often used in close proximity to the tradeoff as ambiguous evidence for it.
- (2) **The bias-variance tradeoff should not be assumed to be universal.** There is evidence that bias and variance trade off in certain methods (e.g. KNN) when varying the right parameter ([Section 3.4.1.1](#)), but there are also counterexamples. For example, there are clear examples of a lack of a bias-variance tradeoff in neural networks ([Chapter 5](#)) and potentially other methods such as decision trees ([Belkin et al., 2019a](#)).
- (3) It should be emphasized that the **PAC upper bounds on test error can be very loose** for the problems we care about in practice (see [Sections 2.3](#) and [3.4.1.2](#) for examples of upper bounds on test error and estimation error). Because these upper bounds are so loose, their qualitative trend (e.g. as number of parameters increases)

is not necessarily an accurate reflection of the qualitative trend of the test error in practice.

Chapter 5

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

by

Brady Neal¹, Sarthak Mittal¹, Aristide Baratin¹, Vinayak Tantia¹,
Matthew Scicluna¹, Simon Lacoste-Julien¹, and Ioannis Mitliagkas¹

(¹) Mila - Quebec AI Institute

ABSTRACT. The bias-variance tradeoff tells us that as model complexity increases, bias falls and variance increases, leading to a U-shaped test error curve. However, recent empirical results with over-parameterized neural networks are marked by a striking absence of the classic U-shaped test error curve: test error keeps decreasing in wider networks. This suggests that there might not be a bias-variance tradeoff in neural networks with respect to network width, unlike was originally claimed by, e.g., Geman et al. (1992). Motivated by the shaky evidence used to support this claim in neural networks, we measure bias and variance in the modern setting. We find that *both* bias *and* variance can decrease as the number of parameters grows. To better understand this, we introduce a new decomposition of the variance to disentangle the effects of optimization and data sampling. We also provide theoretical analysis in a simplified setting that is consistent with our empirical findings.

Keywords: bias-variance tradeoff, neural networks, over-parameterization, generalization

RÉSUMÉ. Le compromis biais-variance nous indique qu'à mesure que la complexité du modèle augmente, le biais diminue et la variance augmente, ce qui conduit à une courbe d'erreur de test en forme de U. Cependant, les résultats empiriques récents avec des réseaux neuronaux sur-paramétrés sont marqués par une absence frappante de la courbe d'erreur de test classique en forme de U : l'erreur de test continue de diminuer dans les réseaux plus larges. Cela donne à penser qu'il n'y a peut-être pas de compromis sur la variance de biais dans les réseaux de neurones en ce qui concerne la largeur du réseau, contrairement à ce que prétendaient, à l'origine, par exemple, Geman et al. (1992). Motivés par les preuves incertaines utilisées à l'appui de cette affirmation dans les réseaux de neurones, nous mesurons les biais et la variance dans le contexte moderne. Nous constatons que le biais d'accentuation et la variance peuvent diminuer à mesure que le nombre de paramètres augmente. Pour mieux comprendre cela, nous introduisons une nouvelle décomposition de la variance pour démêler les effets de l'optimisation et de l'échantillonnage des données. Nous fournissons également une analyse théorique dans un cadre simplifié qui est conforme à nos constatations empiriques.

Mots clés : compromis biais-variance, réseaux de neurones, sur-paramétrage, généralisation

5.1. Introduction

There is a dominant dogma in machine learning:

«*The price to pay for achieving low bias is high variance (Geman et al., 1992).* »

The quantities of interest here are the bias and variance of a learned model's *prediction* on a new input, where the randomness comes from the sampling of the training data. This idea that bias decreases while variance increases with model capacity, leading to a U-shaped test error curve is commonly known as the *bias-variance tradeoff* (Figure 5.1 (left)).

There exist experimental evidence and theory that support the idea of a tradeoff. In their landmark paper, Geman et al. (1992) measure bias and variance in various models. They show convincing experimental evidence for the bias-variance tradeoff in nonparametric methods such as kNN (k-nearest neighbor) and kernel regression. They also show experiments on neural networks and claim that bias decreases and variance increases with network width. Statistical learning theory (Vapnik, 1998) successfully predicts these U-shaped test error curves implied by a tradeoff for a number of classic machine learning models. A key element is identifying a notion of model capacity, understood as the main parameter controlling this tradeoff.

Surprisingly, there is a growing amount of empirical evidence that *wider* networks generalize *better* than their smaller counterparts (Neyshabur et al., 2015; Zagoruyko & Komodakis, 2016; Novak et al., 2018; Lee et al., 2018; Belkin et al., 2019a; Spigler et al., 2018; Liang et al., 2017; Canziani et al., 2016). In those cases the classic U-shaped test error curve is not observed.

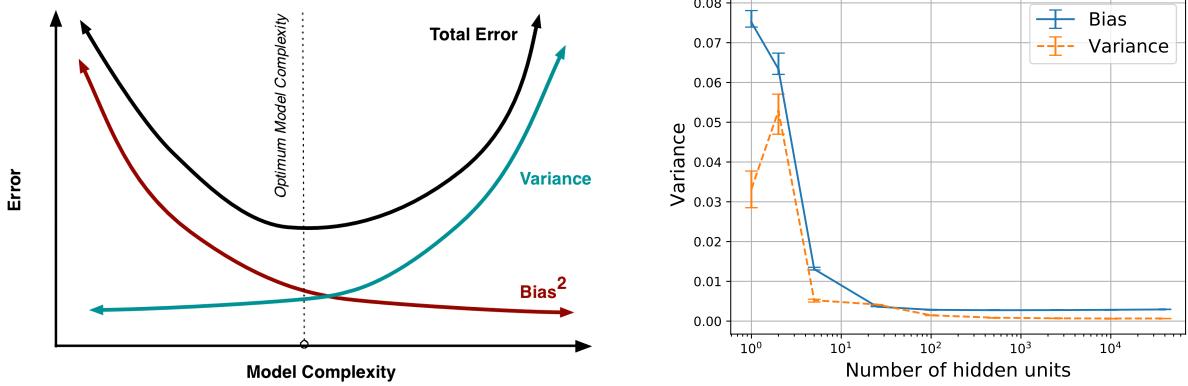


Figure 5.1. On the left is an illustration of the common intuition for the bias-variance tradeoff (Fortmann-Roe, 2012). We find that *both* bias and variance decrease when we increase network width on MNIST (right) and other datasets (Section 5.4). These results seem to contradict the traditional intuition of a strict tradeoff.

A number of different research directions have spawned in response to these findings. Neyshabur et al. (2015) hypothesize the existence of an implicit regularization mechanism. Some study the role that optimization plays (Soudry et al., 2018; Gunasekar et al., 2018). Others suggest new measures of capacity (Liang et al., 2017; Neyshabur et al., 2019). All approaches focus on test error, rather than studying bias and variance directly (Neyshabur et al., 2019; Geiger et al., 2019a; Liang et al., 2017; Belkin et al., 2019a).

Test error analysis does not give a definitive answer on the lack of a bias-variance tradeoff. Consider boosting: it is known that its test error often decreases with the number of rounds (Schapire & Singer, 1999, Figures 8-10). In spite of this monotonicity in test error, Bühlmann & Yu (2003) show that variance grows at an exponentially decaying rate, calling this an “exponential bias-variance tradeoff” (see Section 4.3). To study the bias-variance tradeoff, one has to isolate and measure bias and variance individually. To the best of our knowledge, there has not been published work reporting such measurements on neural networks since Geman et al. (1992).

We go back to basics and study bias and variance. We start by taking a closer look at Geman et al. (1992, Figure 16 and Figure 8 (top))’s experiments with neural networks. We notice that their experiments do not support their claim that “bias falls and variance increases with the number of hidden units.” The authors attribute this inconsistency to convergence issues and maintain their claim that the bias-variance tradeoff is universal. Motivated by this inconsistency, we perform a set of bias-variance experiments with modern neural networks.

We measure prediction bias and variance of fully connected neural networks. These measurements allow us to reason directly about whether there exists a tradeoff with respect to network width. We find evidence that *both* bias *and* variance can decrease at the same

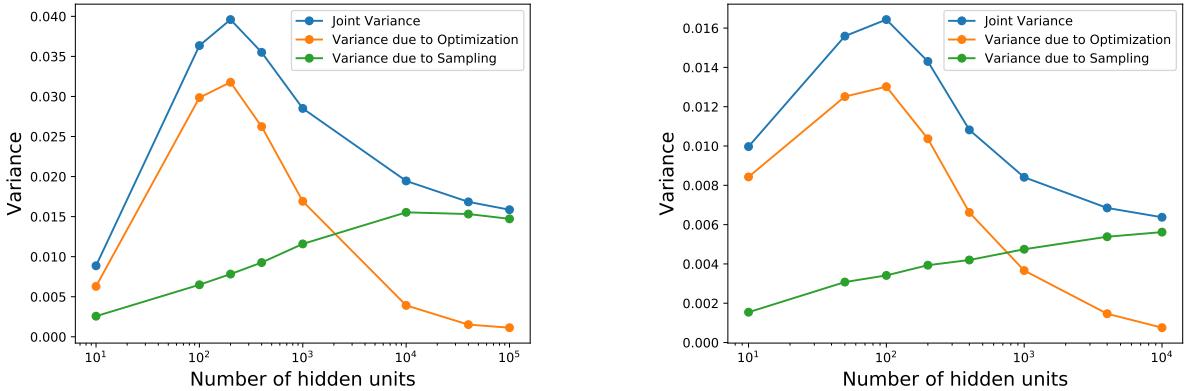


Figure 5.2. Trends of variance due to sampling and variance due to optimization with width on CIFAR10 (left) and on SVHN (right). Variance due to optimization decreases with width, once in the over-parameterized setting. Variance due to sampling plateaus and remains constant. This is in contrast with what the bias-variance tradeoff would suggest.

time as network width increases in common classification and regression settings (Figure 5.1 and Section 5.4).

We observe the qualitative lack of a bias-variance tradeoff in network width with a number of gradient-based optimizers. In order to take a closer look at the roles of optimization and data sampling, we propose a simple decomposition of total prediction variance (Section 5.3.3). We use the law of total variance to get a term that corresponds to average (over data samplings) variance due to optimization and a term that corresponds to variance due to training set sampling of an ensemble of differently initialized networks. **Variance due to optimization is significant in the under-parameterized regime and monotonically decreases with width in the over-parameterized regime.** There, total variance is much lower and dominated by variance due to sampling (Figure 5.2).

We provide theoretical analysis, consistent with our empirical findings, in simplified analysis settings: i) prediction variance does not grow arbitrarily with number of parameters in fixed-design linear models; ii) variance due to optimization diminishes with number of parameters in neural networks under strong assumptions.

Organization. The rest of this paper is organized as follows. We discuss relevant related work in Section 5.2. Section 5.3 establishes necessary preliminaries, including our variance decomposition. In Section 5.4, we empirically study the impact of network width on variance. In Section 5.5, we present theoretical analysis in support of our findings.

5.2. Related work

Neyshabur et al. (2015); Neyshabur (2017) point out that because increasing network width does not lead to a U-shaped test error curve, there must be some form of implicit

regularization controlling capacity. Our work is consistent with this finding, but by approaching the problem from the bias-variance perspective, we gain additional insights: 1) We specifically address the hypothesis that decreased bias must come at the expense of increased variance (see Geman et al. (1992) and Appendix E) by measuring both quantities. 2) Our more fine-grain approach reveals that variance due to optimization vanishes with width, while variances due to sampling increases and levels off. This insight about variance due to sampling is consistent with existing variance results for boosting (Bühlmann & Yu, 2003). To ensure that we are studying networks of increasing capacity, one of the experimental controls we use throughout the paper is to verify that bias is decreasing.

In independent concurrent work, Spigler et al. (2018); Belkin et al. (2019a) point out that generalization error acts according to conventional wisdom in the under-parameterized setting, that it decreases with capacity in the over-parameterized setting, and that there is a sharp transition between the two settings. Although the phrase “bias-variance trade-off” appears in Belkin et al. (2019a)’s title, their work really focuses on the shape of the test error curve: they argue it is not the simple U-shaped curve that conventional wisdom would suggest, and it is not the decreasing curve that Neyshabur et al. (2015) found; it is “double descent curve,” which is essentially a concatenation of the two curves. This is in contrast to our work, where we actually measure bias, variance, and components of variance in this over-parameterized regime. Interestingly, Belkin et al. (2019a)’s empirical study of test error provides some evidence that our bias-variance finding might not be unique to neural networks and might be found in other models such as decision trees.

In subsequent work,¹ Belkin et al. (2019b); Hastie et al. (2019) perform a theoretical analysis of student-teacher linear models (with random features), showing the double descent curve theoretically. Advani & Saxe (2017) also performed a similar analysis. Hastie et al. (2019) is the only one to theoretically analyze variance. Their work differs from ours in that we run experiments with neural networks on complex, real data, while they carry out a theoretical analysis of linear models in a simplified teacher (data generating distribution) setting.

5.3. Preliminaries

5.3.1. Set-up

We consider the typical supervised learning task of predicting an output $y \in \mathcal{Y}$ from an input $x \in \mathcal{X}$, where the pairs (x, y) are drawn from some unknown joint distribution, \mathcal{D} . The learning problem consists of learning a function $h_S : \mathcal{X} \rightarrow \mathcal{Y}$ from a finite training dataset S of m i.i.d. samples from \mathcal{D} . The quality of a predictor h can be quantified by the expected

1. By “subsequent work,” we mean work that appeared on arXiv five months after our paper appeared on arXiv.

error,

$$\mathcal{E}(h) = \mathbb{E}_{(x,y) \sim \mathcal{D}} \ell(h(x), y), \quad (5.3.1)$$

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

In this paper, predictors h_θ are parameterized by the weights $\theta \in \mathbb{R}^N$ of neural networks. We consider the average performance over possible training sets (denoted by the random variable S) of size m . This is the same quantity [Geman et al. \(1992\)](#) consider. While S is the only random quantity studied in the traditional bias-variance decomposition, we also study randomness coming from optimization. We denote the random variable for optimization randomness (e.g. initialization) by O .

Formally, given a fixed training set S and fixed optimization randomness O , the learning algorithm \mathcal{A} produces $\theta = \mathcal{A}(S, O)$. Randomness in optimization translates to randomness in $\mathcal{A}(S, \cdot)$. Given a fixed training set, we encode the randomness due to O in a conditional distribution $p(\theta|S)$. Marginalizing over the training set S of size m gives a marginal distribution $p(\theta) = \mathbb{E}_S p(\theta|S)$ on the weights learned by \mathcal{A} from m samples. In this context, the average performance of the learning algorithm using training sets of size m can be expressed in the following ways:

$$\mathcal{R}_m = \mathbb{E}_{\theta \sim p} \mathcal{E}(h_\theta) = \mathbb{E}_S \mathbb{E}_{\theta \sim p(\cdot|S)} \mathcal{E}(h_\theta) = \mathbb{E}_S \mathbb{E}_O \mathcal{E}(h_\theta) \quad (5.3.2)$$

5.3.2. Bias-variance decomposition

We briefly recall the standard bias-variance decomposition in the case of squared-loss. We work in the context of classification, where each class $k \in \{1 \cdots K\}$ is represented by a one-hot vector in \mathbb{R}^K . The predictor outputs a score or probability vector in \mathbb{R}^K . In this context, the risk in Equation (5.3.2) decomposes into three sources of error ([Geman et al., 1992](#)):

$$\mathcal{R}_m = \mathcal{E}_{\text{noise}} + \mathcal{E}_{\text{bias}} + \mathcal{E}_{\text{variance}} \quad (5.3.3)$$

The first term is an intrinsic error term independent of the predictor:

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

The second term is a bias term:

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

where $\bar{y}(x)$ denotes the expectation $\mathbb{E}[y|x]$ of y given x . The third term is the expected variance of the output predictions:

$$\mathcal{E}_{\text{variance}} = \mathbb{E}_x \text{Var}(h_\theta(x)),$$

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

where the expectation over θ can be done as in Equation (5.3.2). Interpreting this bias-variance decomposition as a bias-variance tradeoff is quite pervasive (see, e.g., Hastie et al. (2001, Chapter 2.9), Goodfellow et al. (2016, 5.4.4), Bishop (2006, Chapter 3.2)). It is generally invoked to emphasize that the model selected should be of the complexity that achieves the optimal balance between bias and variance.

Note that risks computed with classification losses (e.g cross-entropy or 0-1 loss) do not have such a clean bias-variance decomposition (Domingos, 2000; James, 2003). However, it is natural to expect that bias and variance are useful indicators of the performance of models that are not assessed with squared error. In fact, we show the classification risk can be bounded as 4 times the regression risk in Appendix D.4. To empirically examine this connection, in all of our graphs that have “test error” or “training error” on some classification task, we plot the 0-1 classification error (see, e.g., Figure 5.3(b)).

5.3.3. Further decomposing variance into its sources

In the set-up of Section 5.3.1 the prediction is a random variable that depends on two sources of randomness: the randomly drawn training set, S , and any optimization randomness, O , encoded into the conditional $p(\cdot|S)$. In certain regimes, one gets significantly different predictions when using a different initialization. Similarly, the output of a learned predictor changes when we use a different training set. How do we start disentangling variance caused by sampling from variance caused by optimization? There are few different ways; here we describe one of them.

Our goal is to measure prediction variance due to sampling, while controlling for the effect of optimization randomness.

Definition 1 ((Ensemble) Variance due to sampling). *We consider the variance of an ensemble of infinitely many predictors with different optimization randomness (e.g. random initializations):*

$$\text{Var}_S (\mathbb{E}_O [h_\theta(x)|S]).$$

A common practice to estimate variance due to optimization effects is to run multiple seeds on a fixed training set.

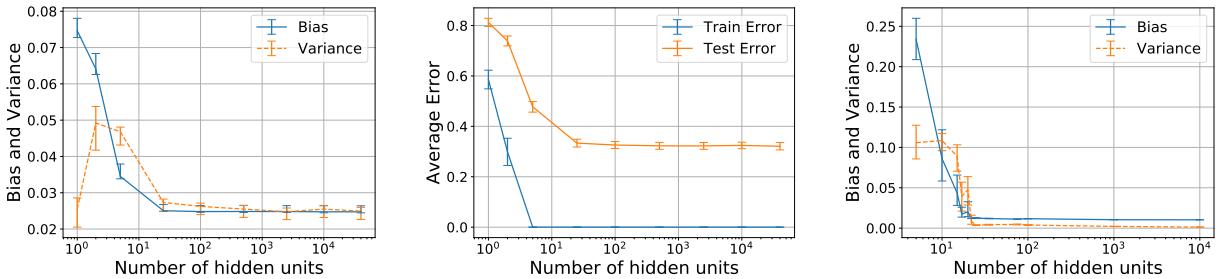
Definition 2 ((Mean) Variance due to optimization). *We consider the average (over training sets) variance over optimization randomness for a fixed training set:*

$$\mathbb{E}_S [\text{Var}_O (h_\theta(x)|S)].$$

The law of total variance naturally decomposes variance into these very terms:

$$\text{Var}(h_\theta(x)) = \mathbb{E}_S [\text{Var}_O (h_\theta(x)|S)] + \text{Var}_S (\mathbb{E}_O [h_\theta(x)|S]) \quad (5.3.4)$$

We use this decomposition to get a finer understanding of our observations (Figure 5.2).



(a) Variance decreases with width, even in the small MNIST setting. (b) Test error trend is same as bias-variance trend (small MNIST). (c) Similar bias-variance trends on sinusoid regression task.

Figure 5.3. We see the same bias-variance trends in small data settings: small MNIST (left) and a regression setting (right).

5.4. Experiments

In this section, we study how variance of fully connected single hidden layer networks varies with width. We provide evidence against Geman et al. (1992)'s important claim about neural networks:

«“The basic trend is what we expect: bias falls and variance increases with the number of hidden units.”»

Our main finding is that, for all tasks that we study, bias and variance both decrease as we scale network width. We also provide a meaningful decomposition of prediction variance into a variance due to sampling term and a variance due to optimization term.

5.4.1. Common experimental details

We run experiments on different datasets: MNIST, SVHN, CIFAR10, small MNIST, and a sinusoid regression task. Averages over data samples are performed by taking the training set S and creating 50 bootstrap replicate training sets S' by sampling with replacement from S . We train 50 different neural networks for each hidden layer size using these different training sets. Then, we estimate $\mathcal{E}_{\text{bias}}$ ² and $\mathcal{E}_{\text{variance}}$ as in Section 5.3.2, where the population expectation \mathbb{E}_x is estimated with an average over the test set. To estimate the two terms from the law of total variance (Equation 5.3.4), we use 10 random seeds for the outer expectation and 10 for the inner expectation, resulting in a total of 100 neural networks for each hidden layer size. Furthermore, we compute 99% confidence intervals for our bias and variance estimates using the bootstrap (Efron, 1979).

The networks are initialized using PyTorch's default initialization, which scales the variance of the weight initialization distribution inversely proportional to the width (LeCun et al., 1998; Glorot & Bengio, 2010). The networks are trained using SGD with momentum and generally run for long after 100% training set accuracy is reached (e.g. 500 epochs for

2. Because we do not have access to \bar{y} , we use the labels y to estimate $\mathcal{E}_{\text{bias}}$. This is equivalent to assuming noiseless labels and is standard procedure for estimating bias (Kohavi & Wolpert, 1996; Domingos, 2000).

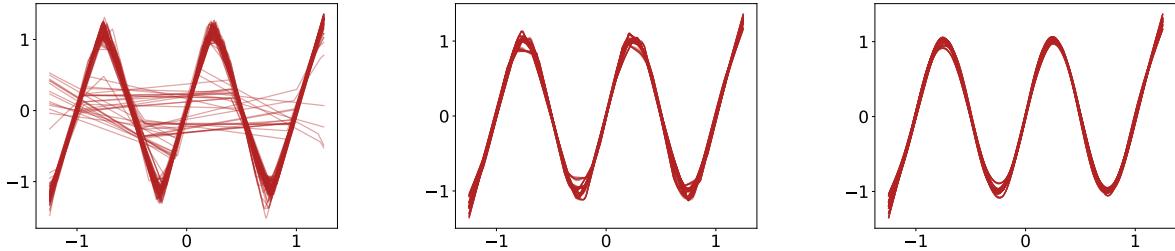


Figure 5.4. Visualization of the 100 different learned functions of single hidden layer neural networks of widths 15, 1000, and 10000 (from left to right) on the task of learning a sinusoid. The learned functions are increasingly similar with width, suggesting decreasing variance. More in [Appendix B.7](#).

full data MNIST and 10000 epochs for small data MNIST). The overall trends we find are robust to how long the networks are trained after the training error converges. The step size hyperparameter is specified in each of the sections, and the momentum hyperparameter is always set to 0.9. To make our study as general as possible, we consider networks without regularization bells and whistles such as weight decay, dropout, or data augmentation, which [Zhang et al. \(2017\)](#) found to not be necessary for good generalization.

5.4.2. Decreasing variance in full data setting

We find a clear decreasing trend in variance with width of the network in the full data MNIST setting ([Figure 5.1](#)). We also see the same trend with CIFAR10 ([Appendix B.1](#)) and SVHN ([Appendix B.2](#)). In these experiments, the same step size is used for all networks for a given dataset (0.1 for MNIST and 0.005 for CIFAR10 and SVHN). The trend is the same with or without early stopping, so early stopping is not necessary to see decreasing variance, similar to how it was not necessary to see better test set performance with width in [Neyshabur et al. \(2015\)](#). Wider ResNets are known to achieve lower test error ([Zagoruyko & Komodakis, 2016](#)); this likely translates to decreasing variance with width in convolutional networks as well. Much of the over-parameterization literature focuses on over-parameterization in width; interestingly, the variance trend is not the same when varying depth ([Appendix C](#)).

5.4.3. Testing the limits: decreasing variance in the small data setting

Decreasing the size of the dataset can only increase variance. To study the robustness of the above observation, we decrease the size of the training set to just 100 examples. In this small data setting, somewhat surprisingly, we still see that *both* bias *and* variance decrease with width ([Figure 5.3\(a\)](#)). The test error behaves similarly ([Figure 5.3\(b\)](#)). Because performance is more sensitive to step size in the small data setting, the step size for each network size is tuned using a validation set (see [Appendix B.4](#) for step sizes). The training

for tuning is stopped after 1000 epochs, whereas the training for the final models is stopped after 10000 epochs. Note that because we see decreasing bias with width, effective capacity is, indeed, increasing while variance is decreasing.

One control that motivates the experimental design choice of optimal step size is that it leads to the conventional decreasing bias trend ([Figure 5.3\(a\)](#)) that indicates increasing effective capacity. In fact, in the corresponding experiment where step size is the same 0.01 for all network sizes, we do not see monotonically decreasing bias ([Appendix B.5](#)).

This sensitivity to step size in the small data setting is evidence that we are testing the limits of our hypothesis. By looking at the small data setting, we are able to test our hypothesis when the ratio of size of network to dataset size is quite large, and we still find this decreasing trend in variance ([Figure 5.3\(a\)](#)).

To see how dependent this phenomenon is on SGD, we also run these experiments using batch gradient descent and PyTorch’s version of LBFGS. Interestingly, we find a decreasing variance trend with those optimizers as well. These experiments are included in [Appendix B.6](#).

5.4.4. Decoupling variance due to sampling from variance due to optimization

In order to better understand this variance phenomenon in neural networks, we separate the variance due to sampling from the variance due to optimization, according to the law of total variance ([Equation 5.3.4](#)). Contrary to what traditional bias-variance tradeoff intuition would suggest, we find variance due to sampling increases slowly and levels off, once sufficiently over-parameterized ([Figure 5.2](#)). Furthermore, we find that variance due to optimization decreases with width, causing the total variance to decrease with width ([Figure 5.2](#)).

A body of recent work has provided evidence that over-parameterization (in width) helps gradient descent optimize to global minima in neural networks ([Du et al., 2019; Du & Lee, 2018; Soltanolkotabi et al., 2017; Livni et al., 2014; Zhang et al., 2018](#)). Always reaching a global minimum implies low variance due to optimization on the *training set*. Our observation of decreasing variance on the *test set* shows that the over-parameterization (in width) effect on optimization seems to extend to generalization, on the data sets we consider.

5.4.5. Visualization with regression on sinusoid

We trained different width neural networks on a noisy sinusoidal distribution with 80 independent training examples. This sinusoid regression setting also exhibits the familiar bias-variance trends ([Figure 5.3\(c\)](#)) and trends of the two components of the variance and the test error ([Figure B.14 of Appendix B.7](#)).

Because this setting is low-dimensional, we can visualize the learned functions. The classic caricature of high capacity models is that they fit the training data in a very erratic way

(example in Figure B.10 of Appendix B.7). We find that wider networks learn sinusoidal functions that are much more similar than the functions learned by their narrower counterparts (Figure 5.4). We have analogous plots for all of the other widths and ones that visualize the variance similar to how it is commonly visualized for Gaussian processes in Appendix B.7.

5.5. Discussion and theoretical insights

Our empirical results demonstrate that in the practical setting, variance due to optimization decreases with network width while variance due to sampling increases slowly and levels off once sufficiently over-parameterized. In Section 5.5.1, we discuss the simple case of linear models and point out that non-increasing variance can already be seen in the over-parameterized setting. In Section 5.5.2 we take inspiration from linear models to provide arguments for the behavior of variance in increasingly wide neural networks, and we discuss the assumptions we make.

5.5.1. Insights from linear models

In this section, we review the classic result that the variance of a linear model grows with the number of parameters (Hastie et al., 2009, Section 7.3) and point out that variance behaves differently in the over-parameterized setting.

We consider least-squares linear regression in a standard setting which assumes a noisy linear mapping $y = \theta^T x + \epsilon$ between input feature vectors $x \in \mathbb{R}^N$ and real outputs, where ϵ denotes the noise random variable with $\mathbb{E}[\epsilon] = 0$ and $\text{Var}(\epsilon) = \sigma_\epsilon^2$. In this context, the over-parameterized setting is when the dimension N of the input space is larger than the number m of examples.

Let X denote the $m \times N$ design matrix whose i^{th} row is the training point x_i^T , let Y denote the corresponding labels, and let $\Sigma = X^T X$ denote the empirical covariance matrix. We consider the fixed-design setting where X is fixed, so all of the randomness due to data sampling comes solely from ϵ . \mathcal{A} learns weights $\hat{\theta}$ from (X, Y) , either by a closed-form solution or by gradient descent, using a standard initialization $\theta_0 \sim \mathcal{N}(0, \frac{1}{N}I)$. The predictor makes a prediction on $x \sim \mathcal{D}$: $h(x) = \hat{\theta}^T x$. Then, the quantity we care about is $\mathbb{E}_x \text{Var}(h(x))$.

5.5.1.1. Under-parameterized setting

The case where $N \leq m$ is standard: if X has maximal rank, Σ is invertible; the solution is independent of the initialization and given by $\hat{\theta} = \Sigma^{-1} X^T Y$. All of the variance is a result of randomness in the noise ϵ . For a fixed x ,

$$\text{Var}(h(x)) = \sigma_\epsilon^2 \text{Tr}(x x^T \Sigma^{-1}). \quad (5.5.1)$$

This grows with the number of parameters N . For example, taking the expected value over the empirical distribution, \hat{p} , of the sample, we recover that the variance grows with N :

$$\mathbb{E}_{x \sim \hat{p}}[\text{Var}(h(x))] = \frac{N}{m} \sigma_\epsilon^2. \quad (5.5.2)$$

We provide a reproduction of the proofs in [Appendix D.1](#).

5.5.1.2. Over-parameterized setting

The over-parameterized case where $N > m$ is more interesting: even if X has maximal rank, Σ is not invertible. This leads to a subspace of solutions, but gradient descent yields a unique solution from updates that belong to the span of the training points x_i (row space of X) ([LeCun et al., 1991](#)), which is of dimension $r = \text{rank}(X) = \text{rank}(\Sigma)$. Correspondingly, no learning occurs in the null space of X , which is of dimension $N - r$. Therefore, gradient descent yields the solution that is closest to initialization: $\hat{\theta} = P_\perp(\theta_0) + \Sigma^+ X^T Y$, where P_\perp projects onto the null space of X and $+$ denotes the Moore-Penrose inverse.

The variance has two contributions: one due to initialization and one due to sampling (here, the noise ϵ), as in [Equation \(5.3.4\)](#). These are made explicit in [Proposition 5.5.1](#).

Proposition 5.5.1 (Variance in over-parameterized linear models). *Consider the over-parameterized setting where $N > m$. For a fixed x , the variance decomposition of [Equation \(5.3.4\)](#) yields*

$$\text{Var}(h(x)) = \frac{1}{N} \|P_\perp(x)\|^2 + \sigma_\epsilon^2 \text{Tr}(xx^T \Sigma^+). \quad (5.5.3)$$

This does not grow with the number of parameters N . In fact, because Σ^{-1} is replaced with Σ^+ , the variance *scales as the dimension of the data* (i.e the rank of X), as opposed to the number of parameters. For example, taking the expected value over the empirical distribution, \hat{p} , of the sample, we obtain

$$\mathbb{E}_{x \sim \hat{p}}[\text{Var}(h(x))] = \frac{r}{m} \sigma_\epsilon^2, \quad (5.5.4)$$

where $r = \text{rank}(X)$. We provide the proofs for over-parameterized linear models in [Appendix D.2](#).

5.5.2. A more general result

We will illustrate our arguments in the following simplified setting, where \mathcal{M} , \mathcal{M}^\perp , and $d(N)$ are the more general analogs of $\text{rowspace}(X)$, $\text{nullspace}(X)$, and r (respectively):

Setting. Let N be the dimension of the parameter space. The prediction for a fixed example x , given by a trained network parameterized by θ depends on:

- (i) a subspace of the parameter space, $\mathcal{M} \in \mathbb{R}^N$ with relatively small dimension, $d(N)$, which depends only on the learning task.

(ii) parameter components corresponding to directions orthogonal to \mathcal{M} . The orthogonal \mathcal{M}^\perp of \mathcal{M} has dimension, $N - d(N)$, and is essentially irrelevant to the learning task.

We can write the parameter vector as a sum of these two components $\theta = \theta_{\mathcal{M}} + \theta_{\mathcal{M}^\perp}$. We will further make the following 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}).

5.5.2.1. Variance due to initialization

Given the above assumptions, the following result shows that the variance from initialization³ vanishes as we increase N . The full proof, which builds on concentration results for Gaussians (based on Levy’s lemma (Ledoux, 2001)), is given in [Appendix D.3](#).

Theorem 5.5.2 (Decay of variance due to initialization). *Consider the setting of Section 5.5.2 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,*

$$\text{Var}_{\theta_0}(h_\theta(x)) \leq C \frac{2L^2}{N} \quad (5.5.5)$$

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

This result guarantees that the variance decreases to zero as N increases, provided the Lipschitz constant L grows more slowly than the square root of dimension, $L = o(\sqrt{N})$.

5.5.2.2. Variance due to sampling

Under the above assumptions, the parameters at the end of learning take the form $\theta = \theta_{\mathcal{M}}^* + \theta_{0\mathcal{M}^\perp}$. For fixed initialization, the only source of variance of the prediction is the randomness of $\theta_{\mathcal{M}}^*$ on the learning manifold. The variance depends on the parameter dimensionality only through $\dim \mathcal{M} = d(N)$, and hence remains constant if $d(N)$ does (see [Li et al. \(2018\)](#)’s “intrinsic dimension”).

Discussion on assumptions. We made strong assumptions, but there is some support for them in the literature. The existence of a subspace \mathcal{M}_\perp in which no learning occurs was also conjectured by [Advani & Saxe \(2017\)](#) and shown to hold in linear neural networks under a simplifying assumption that decouples the dynamics of the weights in different layers. [Li et al. \(2018\)](#) empirically showed the existence of a critical number $d(N) = d$ of relevant parameters for a given learning task, independent of the size of the model. [Sagun et al. \(2017\)](#) showed that the spectrum of the Hessian for over-parameterized networks splits into

3. Among the different sources of optimization randomness, we focus on randomness from initialization and do not focus on randomness from stochastic mini-batching because we found the phenomenon of decreasing variance with width persists when using *batch* gradient descent ([Section 5.4.3, Appendix B.6](#)).

(*i*) a bulk centered near zero and (*ii*) a small number of large eigenvalues; and Gur-Ari et al. (2018) recently gave evidence that the small subspace spanned by the Hessian’s top eigenvectors is preserved over long periods of training. These results suggest that learning occurs mainly in a small number of directions.

5.6. Conclusion and future work

We provide evidence against Geman et al. (1992)’s claim that “the price to pay for achieving low bias is high variance,” finding that *both* bias *and* variance decrease with network width. Geman et al. (1992)’s claim is found throughout machine learning and is meant to generally apply to all of machine learning (Appendix E), and it is correct in many cases (e.g. kNN, kernel regression, splines). Is this lack of a tradeoff specific to neural networks or is it present in other models as well such as decision trees?

We propose a new decomposition of the variance, finding variance due to sampling (analog of regular variance in simple settings) does not appear to be dependent on width, once sufficiently over-parameterized, and that variance due to optimization decreases with width. By taking inspiration from linear models, we perform a theoretical analysis of the variance that is consistent with our empirical observations.

We view future work that uses the bias-variance lens as promising. For example, a probabilistic notion of effective capacity of a model is natural when studying generalization through this lens (Appendix A). We did not study how bias and variance change over the course of training; that would make an interesting direction for future work. We also see further theoretical treatment of variance as a fruitful direction for better understanding complexity and generalization abilities of neural networks.

Acknowledgments

We thank Yoshua Bengio, Lechao Xiao, Aaron Courville, Sharan Vaswani, Roman Novak, Xavier Bouthillier, Stanislaw Jastrzebski, Gaetan Marceau Caron, Rémi Le Priol, Guillaume Lajoie, and Joseph Cohen for helpful discussions. Additionally, we thank SigOpt for access to their professional hyperparameter tuning services. This research was partially supported by the NSERC Discovery Grant (RGPIN-2017-06936 and RGPIN-2019-06512), by a Google Focused Research Award, the FRQNT nouveaux chercheurs program (2019-NC-257943), a startup grant by IVADO and the Canada CIFAR AI chair program. We thank NVIDIA for donating a DGX-1 computer used in this work.

Chapter 6

Conclusion and Discussion

It is time that the bias-variance tradeoff sections of textbooks are updated. We reviewed the history of the bias-variance tradeoff (including evidence for it) and its prevalence in textbooks in [Section 3.4](#). We refuted [Geman et al. \(1992\)](#)'s influential claims in [Section 4.1](#) by referencing recent measurements of bias and variance in neural networks ([Chapter 5](#), [Neal et al. \(2018\)](#)). We covered the emerging alternative hypothesis that the test error (risk) actually follows a “double descent” curve (as opposed to a U-shaped curve) in [Section 4.4](#). Finally, we suggested specific changes to the bias-variance tradeoff section of textbooks in [Section 4.5](#).

The specific changes can be simple qualifications. For example, the bias-variance decomposition ([Section 3.3](#)) is often used as evidence for the bias-variance tradeoff. However, this is misleading, unless one assumes fixed risk, which is often not the case when increasing model complexity. Additionally, though there seems to be clear evidence for the bias-variance tradeoff in many nonparametric methods ([Section 3.4.1.1](#)), we should not generalize this to all learning algorithms and assume that the bias-variance tradeoff is universal. These points can be easily clarified in teaching by clearly pointing out that the bias-variance decomposition does not imply a tradeoff and that seeing a tradeoff in certain models does not mean that we see a tradeoff in other models.

It remains to be seen whether there is a general shape of the risk curve that we should expect for all models as we increase model complexity. For now, it appears that there is not: we see clear U-shaped curves in some nonparametric methods such as KNN, kernel regression, and splines ([Section 3.4.1.1](#)), and we see clear double descent curves in neural networks ([Nakkiran et al., 2019](#); [Belkin et al., 2019a](#); [Spigler et al., 2018](#); [Geiger et al., 2019b](#); [Advani & Saxe, 2017](#)), with preliminary evidence of double descent curves in random forests ([Belkin et al., 2019a](#)).

Bibliography

- Abu-Mostafa, Y. S., Magdon-Ismail, M., and Lin, H.-T. *Learning From Data*. AMLBook, 2012.
- Advani, M. S. and Saxe, A. M. High-dimensional dynamics of generalization error in neural networks. *CoRR*, abs/1710.03667, 2017.
- Arpit, D., Jastrzebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., and Lacoste-Julien, S. A closer look at memorization in deep networks. *ICML 2017*, 70:233–242, 06–11 Aug 2017.
- Baldazzi, D., Frean, M., Leary, L., Lewis, J. P., Ma, K. W.-D., and McWilliams, B. The shattered gradients problem: If resnets are the answer, then what is the question? In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 342–350, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.
- Barron, A. R. Approximation and estimation bounds for artificial neural networks. *Machine Learning*, 14(1):115–133, Jan 1994.
- Bartlett, P. L. and Mendelson, S. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, March 2003.
- Bartlett, P. L., Maiorov, V., and Meir, R. Almost linear vc dimension bounds for piecewise polynomial networks. In Kearns, M. J., Solla, S. A., and Cohn, D. A. (eds.), *Advances in Neural Information Processing Systems 11*, pp. 190–196. MIT Press, 1998.
- Baum, E. B. and Haussler, D. What size net gives valid generalization? *Neural Comput.*, 1(1):151–160, March 1989.
- Belkin, M., Hsu, D., Ma, S., and Mandal, S. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019a.
- Belkin, M., Hsu, D., and Xu, J. Two models of double descent for weak features, 2019b.
- Bengio, Y., Simard, P., and Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, March 1994.
- Bishop, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

- Bös, S. and Opper, M. Dynamics of training. In *Advances in Neural Information Processing Systems*, pp. 141–147, 1997.
- Bottou, L. and Bousquet, O. The tradeoffs of large scale learning. In Platt, J. C., Koller, D., Singer, Y., and Roweis, S. T. (eds.), *Advances in Neural Information Processing Systems 20*, pp. 161–168. Curran Associates, Inc., 2008.
- Bousquet, O. and Elisseeff, A. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, March 2002.
- Bühlmann, P. and Yu, B. Boosting with the l_2 loss: regression and classification. *Journal of the American Statistical Association*, 98(462):324–339, 2003.
- Canziani, A., Paszke, A., and Culurciello, E. An analysis of deep neural network models for practical applications. *CoRR*, abs/1605.07678, 2016.
- Cybenko, G. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, Dec 1989.
- Domingos, P. A unified bias-variance decomposition and its applications. In *In Proc. 17th International Conf. on Machine Learning*, pp. 231–238. Morgan Kaufmann, 2000.
- Du, S. and Lee, J. On the power of over-parametrization in neural networks with quadratic activation. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1329–1338, Stockholm Sweden, 10–15 Jul 2018. PMLR.
- Du, S., Zhai, X., Poczos, B., and Singh, A. Gradient descent provably optimizes over-parameterized neural networks. volume abs/1810.02054, 2019.
- Duda, R. O., Hart, P. E., and Stork, D. G. *Pattern Classification*. Wiley, New York, 2 edition, 2001.
- Efron, B. Bootstrap methods: Another look at the jackknife. *Ann. Statist.*, 7(1):1–26, 01 1979.
- EliteDataScience. Wtf is the bias-variance tradeoff? (infographic), May 2018.
- Fortmann-Roe, S. Understanding the bias-variance tradeoff, June 2012.
- Geiger, M., Jacot, A., Spigler, S., Gabriel, F., Sagun, L., d’Ascoli, S., Biroli, G., Hongler, C., and Wyart, M. Scaling description of generalization with number of parameters in deep learning. *CoRR*, abs/1901.01608, 2019a.
- Geiger, M., Spigler, S., d’Ascoli, S., Sagun, L., Baity-Jesi, M., Biroli, G., and Wyart, M. The jamming transition as a paradigm to understand the loss landscape of deep neural networks. *Physical review E*, 100 1-1:012115, 2019b.
- Geman, S., Bienenstock, E., and Doursat, R. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 1992.
- Geurts, P. *Contributions to decision tree induction: bias/variance tradeoff and time series classification*. PhD thesis, University of Liège Belgium, 2002.

- Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Teh, Y. W. and Titterington, M. (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.
- Gonzalez, J. E. Linear regression and the bias variance tradeoff, 2016. lecture notes.
- Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- Grenander, U. On empirical spectral analysis of stochastic processes. *Ark. Mat.*, 1(6):503–531, 08 1952.
- Gunasekar, S., Lee, J., Soudry, D., and Srebro, N. Characterizing implicit bias in terms of optimization geometry. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1832–1841, Stockholm, Sweden, 10–15 Jul 2018. PMLR.
- Gur-Ari, G., Roberts, D. A., and Dyer, E. Gradient descent happens in a tiny subspace. *arXiv preprint arXiv: arXiv:1812.04754*, 2018.
- Guyon, I. *Réseaux de neurones pour la reconnaissance des formes: architectures et apprentissage*. PhD thesis, 1988.
- Harvey, N., Liaw, C., and Mehrabian, A. Nearly-tight VC-dimension bounds for piecewise linear neural networks. In Kale, S. and Shamir, O. (eds.), *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pp. 1064–1068, Amsterdam, Netherlands, 07–10 Jul 2017. PMLR.
- Hastie, T. and Tibshirani, R. *Generalized additive models*. Wiley Online Library, 1990.
- Hastie, T., Tibshirani, R., and Friedman, J. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.
- Hastie, T., Tibshirani, R., and Friedman, J. *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition, 2009.
- Hastie, T., Montanari, A., Rosset, S., and Tibshirani, R. J. Surprises in high-dimensional ridgeless least squares interpolation, 2019.
- He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, June 2016.
- Hochreiter, S. Untersuchungen zu dynamischen neuronalen Netzen. Diploma thesis, Institut für Informatik, Lehrstuhl Prof. Brauer, Technische Universität München, 1991.
- Hornik, K. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251 – 257, 1991.

- Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Bach, F. and Blei, D. (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 448–456, Lille, France, 07–09 Jul 2015. PMLR.
- James, G., Witten, D., Hastie, T., and Tibshirani, R. *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014.
- James, G. M. Variance and bias for general loss functions. In *Machine Learning*, pp. 115–135, 2003.
- Jastrzkebski, S., Kenton, Z., Arpit, D., Ballas, N., Fischer, A., Bengio, Y., and Storkey, A. J. Three factors influencing minima in SGD. *CoRR*, abs/1711.04623, 2017.
- Karpinski, M. and Macintyre, A. Polynomial bounds for vc dimension of sigmoidal neural networks. In *Proceedings of the Twenty-seventh Annual ACM Symposium on Theory of Computing*, STOC ’95, pp. 200–208, New York, NY, USA, 1995. ACM.
- Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*, 2017.
- Kohavi, R. and Wolpert, D. Bias plus variance decomposition for zero-one loss functions. In *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*, ICML’96, pp. 275–283, San Francisco, CA, USA, 1996. Morgan Kaufmann Publishers Inc.
- LeCun, Y., Kanter, I., and Solla, S. Eigenvalues of covariance matrices: Application to neural-network learning. *Physical Review Letters*, 66:2396–2399, 05 1991.
- LeCun, Y., Bottou, L., Orr, G. B., and Müller, K.-R. Efficient backprop. In *Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop*, pp. 9–50, London, UK, UK, 1998. Springer-Verlag.
- Ledoux, M. *The Concentration of Measure Phenomenon*. Mathematical surveys and monographs. American Mathematical Society, 2001.
- Lee, J., Sohl-dickstein, J., Pennington, J., Novak, R., Schoenholz, S., and Bahri, Y. Deep neural networks as gaussian processes. In *International Conference on Learning Representations*, 2018.
- Leshno, M. and Schocken, S. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6:861–867, 1993.
- Li, C., Farkhoor, H., Liu, R., and Yosinski, J. Measuring the intrinsic dimension of objective landscapes. *ICLR 2018*, 2018.
- Liang, T., Poggio, T. A., Rakhlin, A., and Stokes, J. Fisher-rao metric, geometry, and complexity of neural networks. *CoRR*, abs/1711.01530, 2017.
- Livni, R., Shalev-Shwartz, S., and Shamir, O. On the computational efficiency of training neural networks. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and

- Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 27*, pp. 855–863. Curran Associates, Inc., 2014.
- Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of Machine Learning*. The MIT Press, 2012.
- Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., and Sutskever, I. Deep double descent: Where bigger models and more data hurt, 2019.
- Neal, B. and Mitliagkas, I. In support of over-parametrization in deep reinforcement learning: an empirical study. *ICML 2019 Workshop on Identifying and Understanding Deep Learning Phenomena*, 2019.
- Neal, B., Mittal, S., Baratin, A., Tantia, V., Scicluna, M., Lacoste-Julien, S., and Mitliagkas, I. A modern take on the bias-variance tradeoff in neural networks, 2018.
- Neyshabur, B. Implicit regularization in deep learning. *arXiv preprint arXiv:1709.01953*, 2017.
- Neyshabur, B., Tomioka, R., and Srebro, N. In search of the real inductive bias: On the role of implicit regularization in deep learning. *International Conference on Learning Representations workshop track*, 2015.
- Neyshabur, B., Li, Z., Bhojanapalli, S., LeCun, Y., and Srebro, N. The role of over-parametrization in generalization of neural networks. In *International Conference on Learning Representations*, 2019.
- Novak, R., Bahri, Y., Abolafia, D. A., Pennington, J., and Sohl-Dickstein, J. Sensitivity and generalization in neural networks: an empirical study. In *International Conference on Learning Representations*, 2018.
- Opper, M. Statistical mechanics of learning: Generalization. *The Handbook of Brain Theory and Neural Networks*, pp. 922–925, 1995.
- Opper, M. Learning to generalize. *Frontiers of Life*, 3(part 2):763–775, 2001.
- Park, D., Sohl-Dickstein, J., Le, Q., and Smith, S. The effect of network width on stochastic gradient descent and generalization: an empirical study. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5042–5051, Long Beach, California, USA, 09–15 Jun 2019. PMLR.
- Pennington, J., Schoenholz, S., and Ganguli, S. Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 4785–4795. Curran Associates, Inc., 2017.
- Sagun, L., Evci, U., Guney, V. U., Dauphin, Y., and Bottou, L. Empirical analysis of the hessian of over-parametrized neural networks. 2017.
- Saxe, A. M., McClelland, J. L., and Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural network. In *In International Conference on Learning*

- Representations*, 2014.
- Schapire, R. E. and Singer, Y. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, Dec 1999.
- Schoenholz, S. S., Gilmer, J., Ganguli, S., and Sohl-Dickstein, J. Deep information propagation. *ICLR 2017*, 2017.
- Shalev-Shwartz, S. and Ben-David, S. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.
- Smith, S. L., Kindermans, P.-J., and Le, Q. V. Don’t decay the learning rate, increase the batch size. In *International Conference on Learning Representations*, 2018.
- Soltanolkotabi, M., Javanmard, A., and Lee, J. D. Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *CoRR*, abs/1707.04926, 2017.
- Soudry, D., Hoffer, E., and Srebro, N. The implicit bias of gradient descent on separable data. In *International Conference on Learning Representations*, 2018.
- Spigler, S., Geiger, M., d’Ascoli, S., Sagun, L., Biroli, G., and Wyart, M. A jamming transition from under- to over-parametrization affects loss landscape and generalization. *CoRR*, abs/1810.09665, 2018.
- Vapnik, V. N. *Statistical learning theory*. Adaptive and learning systems for signal processing, communications and control series. John Wiley & Sons, New York. A Wiley-Interscience Publication, 1998.
- Vapnik, V. N. An overview of statistical learning theory. *Trans. Neur. Netw.*, 10(5):988–999, September 1999.
- Wahba, G. and Wold, S. A completely automatic french curve: fitting spline functions by cross validation. *Communications in Statistics*, 4(1):1–17, 1975.
- Xiao, L., Bahri, Y., Sohl-Dickstein, J., Schoenholz, S., and Pennington, J. Dynamical isometry and a mean field theory of CNNs: How to train 10,000-layer vanilla convolutional neural networks. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5393–5402, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.
- Zagoruyko, S. and Komodakis, N. Wide residual networks. In Richard C. Wilson, E. R. H. and Smith, W. A. P. (eds.), *Proceedings of the British Machine Vision Conference (BMVC)*, pp. 87.1–87.12. BMVA Press, September 2016.
- Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. *ICLR 2017*, 2017.
- Zhang, C., Liao, Q., Rakhlin, A., Miranda, B., Golowich, N., and Poggio, T. A. Theory of deep learning iib: Optimization properties of SGD. *CoRR*, abs/1801.02254, 2018.

Appendix A

Probabilistic notion of effective capacity

The problem with classical complexity measures is that they do not take into account optimization and have no notion of what will actually be learned. [Arpit et al. \(2017, Section 1\)](#) define a notion of an *effective* hypothesis class to take into account what functions are possible to be learned by the learning algorithm.

However, this still has the problem of not taking into account what hypotheses are *likely* to be learned. To take into account the probabilistic nature of learning, we define the ϵ -*hypothesis class* for a data distribution \mathcal{D} and learning algorithm \mathcal{A} , that contains the hypotheses which are at least ϵ -likely for some $\epsilon > 0$:

$$\mathcal{H}_{\mathcal{D}}(\mathcal{A}) = \{h : p(h(\mathcal{A}, S)) \geq \epsilon\}, \quad (\text{A.0.1})$$

where S is a training set drawn from \mathcal{D}^m , $h(\mathcal{A}, S)$ is a random variable drawn from the distribution over learned functions induced by \mathcal{D} and the randomness in \mathcal{A} ; p is the corresponding density. Thinking about a model's ϵ -hypothesis class can lead to drastically different intuitions for the complexity of a model and its variance ([Figure A.1](#)). This is at the core of the intuition for why the traditional view of bias-variance as a tradeoff does not hold in all cases.

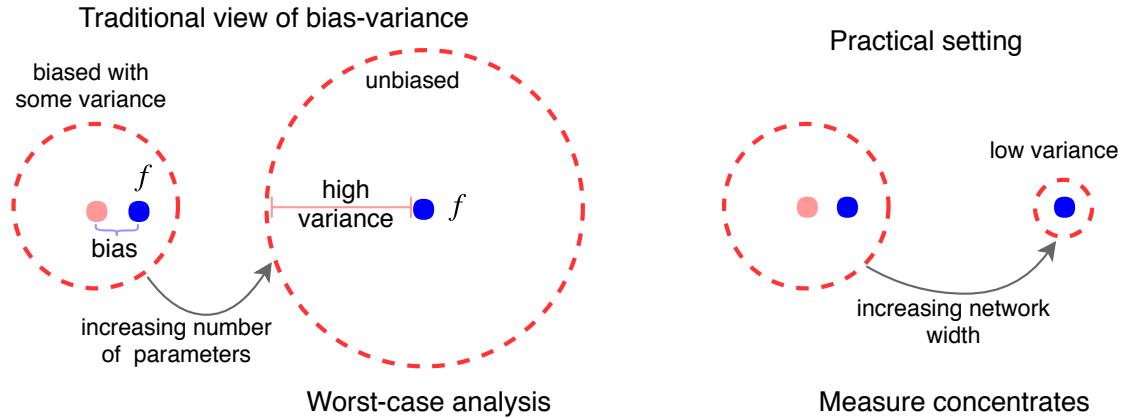


Figure A.1. The dotted red circle depicts a cartoon version of the ϵ -hypothesis class of the learner. The left side reflects common intuition, as informed by the bias-variance tradeoff and worst-case analysis from statistical learning theory. The right side reflects our view that variance can decrease with network width.

Appendix B

Additional empirical results and discussion

B.1. CIFAR10

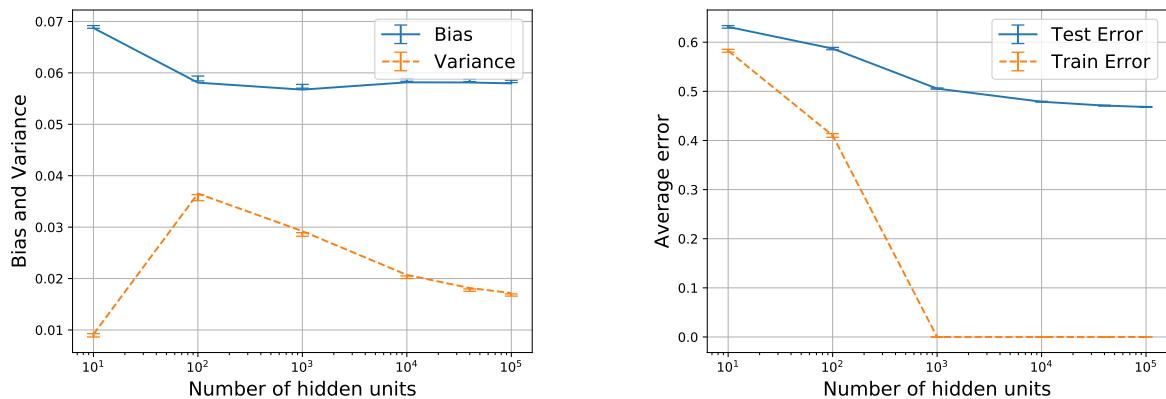


Figure B.1. Bias-variance plot (left) and corresponding train and test error (right) for CIFAR10 after training for 150 epochs with step size 0.005 for all networks.

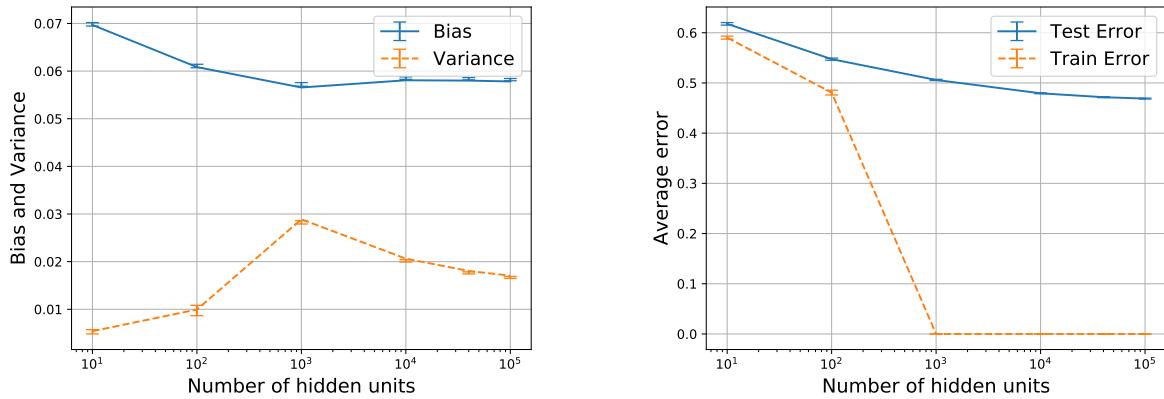


Figure B.2. Bias-variance plot (left) and corresponding train and test error (right) for CIFAR10 after training for using *early stopping* with step size 0.005 for all networks.

B.2. SVHN

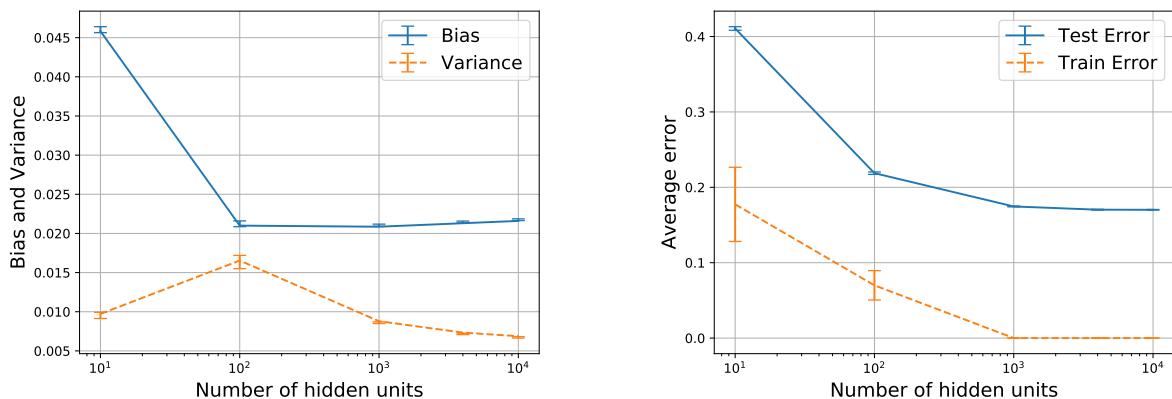


Figure B.3. Bias-variance plot (left) and corresponding train and test error (right) for SVHN after training for 150 epochs with step size 0.005 for all networks.

B.3. MNIST

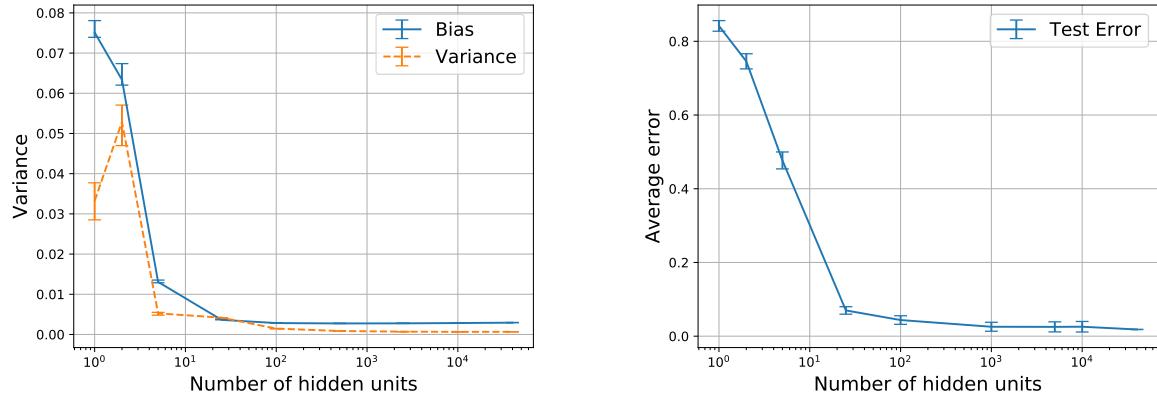


Figure B.4. MNIST bias-variance plot from main paper (left) next to the corresponding test error (right)

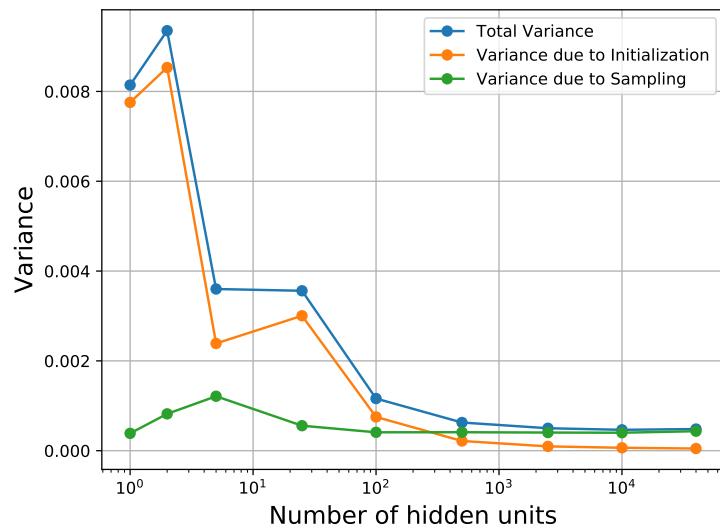
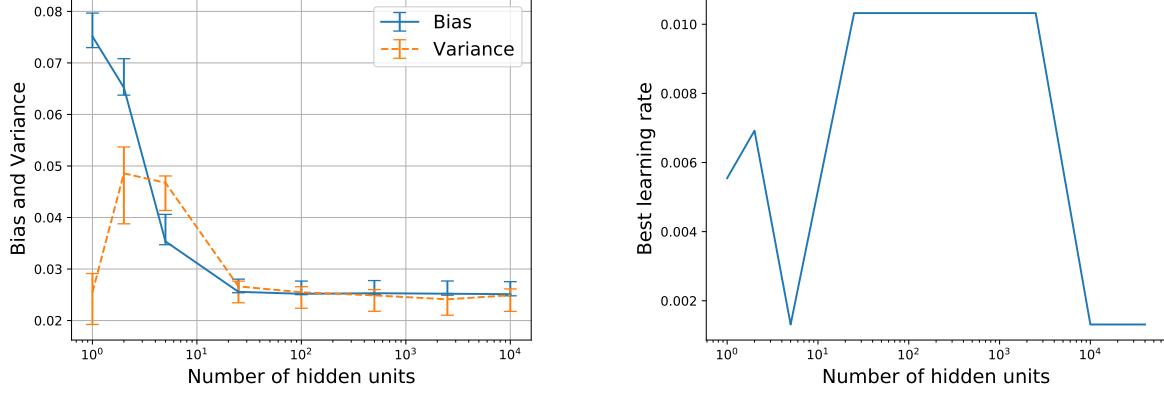


Figure B.5. Decomposed variance on MNIST

B.4. Tuned learning rates for SGD



(a) Variance decreases with width, even in the small data setting (SGD). This figure is in the main paper, but we include it here to compare with the corresponding step sizes used.

(b) Corresponding optimal learning rates found, by random search, and used.

Figure B.6. Tuned learning rates used for small data MNIST

B.5. Fixed learning rate results for small data MNIST

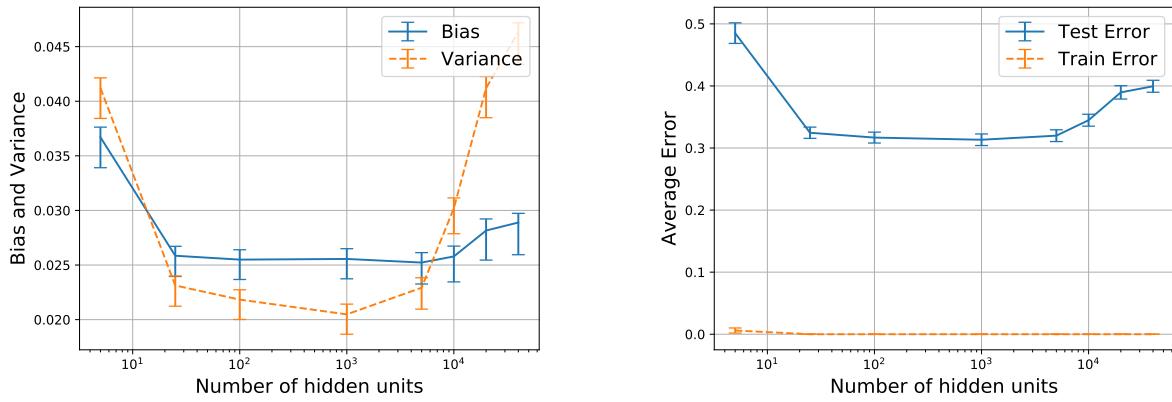


Figure B.7. Variance on small data with a fixed learning rate of 0.01 for all networks.

Note that the U curve shown in Figure B.7 when we do not tune the step size is explained by the fact that the constant step chosen is a “good” step size for some networks and “bad” for others. Results from Keskar et al. (2017), Smith et al. (2018), and Jastrzkebski et al. (2017) show that a step size that corresponds well to the noise structure in SGD is important for achieving good test set accuracy. Because our networks are different sizes, their stochastic

optimization process will have a different landscape and noise structure. By tuning the step size, we are making the experimental design choice to keep *optimality of step size* constant across networks, rather than keeping step size constant across networks. To us, choosing this control makes much more sense than choosing to control for step size. Note that Park et al. (2019) show that, as long as the network size is not too big, controlling for “optimality of step size” and “keeping step size constant” with increasing network width actually correspond to the same thing, as long as the networks are not too big, which fits well with the fact that we used the same step size across network widths for almost all of our experiments.

B.6. Other optimizers for width experiment on small data MNIST

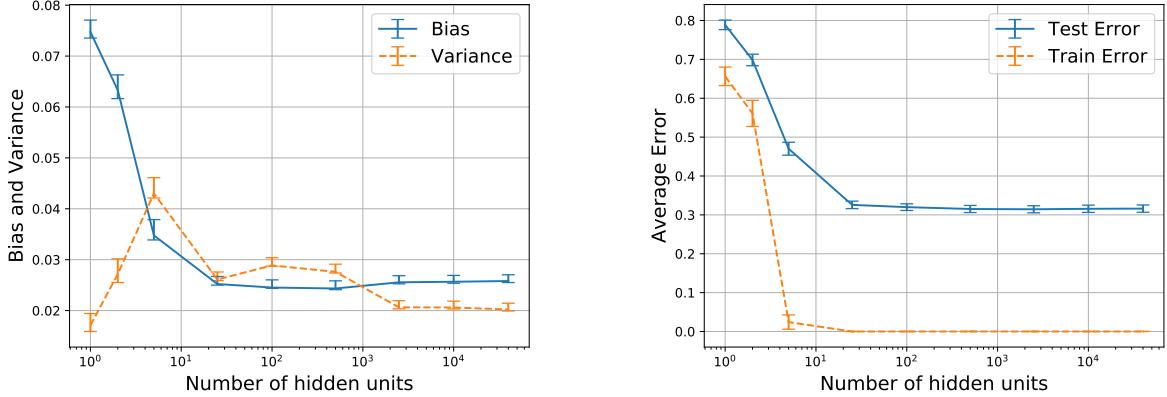


Figure B.8. Variance decreases with width in the small data setting, even when using batch gradient descent.

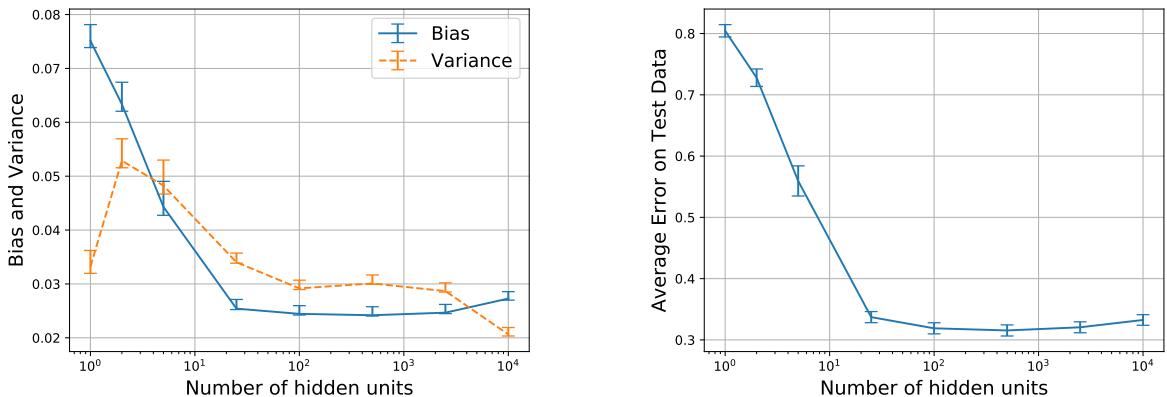
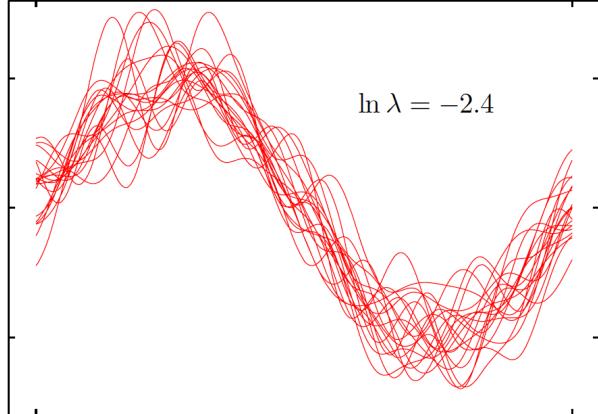
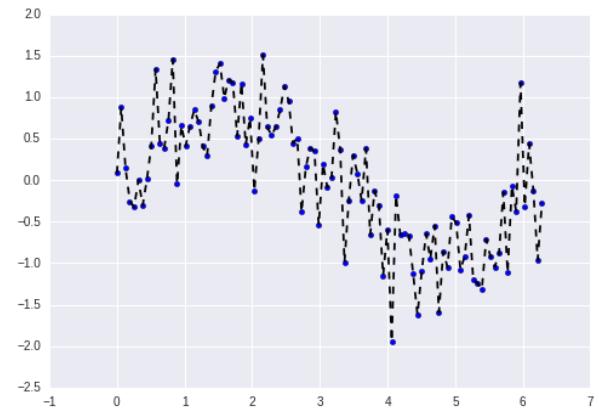


Figure B.9. Variance decreases with width in the small data setting, even when using a strong optimizer, such as PyTorch’s LBFGS, as the optimizer.

B.7. Sinusoid regression experiments



(a) Example of the many different functions learned by a high variance learner ([Bishop, 2006](#))



(b) Caricature of a single function learned by a high variance learner ([EliteDataScience, 2018](#))

Figure B.10. Caricature examples of high variance learners on sinusoid task. Below, we find that this does not happen with increasingly wide neural networks ([Figure B.12](#) and [Figure B.13](#)).

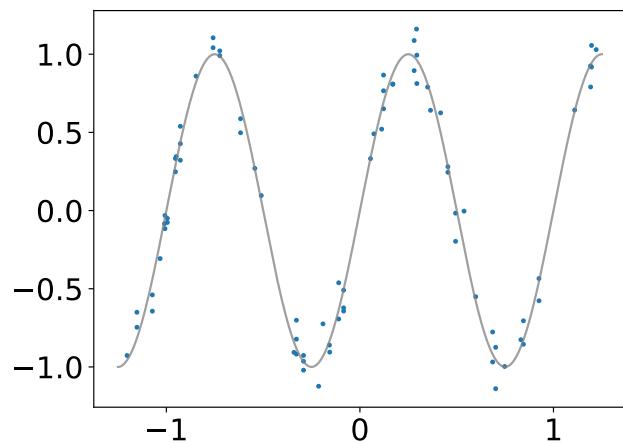


Figure B.11. Target function of the noisy sinusoid regression task (in gray) and an example of a training set (80 data points) sampled from the noisy distribution.

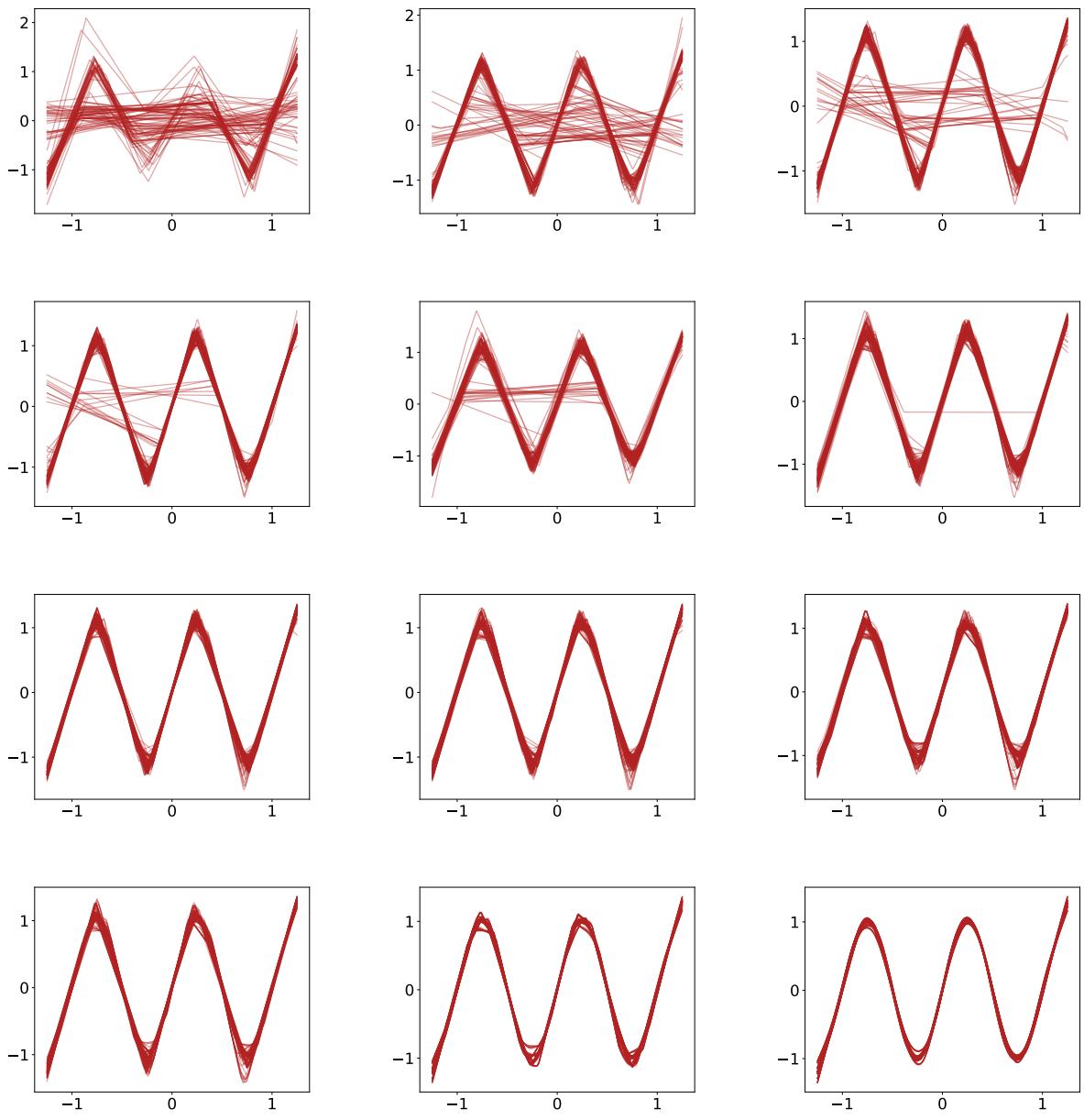


Figure B.12. Visualization of 100 different functions learned by the different width neural networks. Darker color indicates higher density of different functions. Widths in increasing order from left to right and top to bottom: 5, 10, 15, 17, 20, 22, 25, 35, 75, 100, 1000, 10000. We do *not* observe the caricature from Figure B.10 as width is increased.

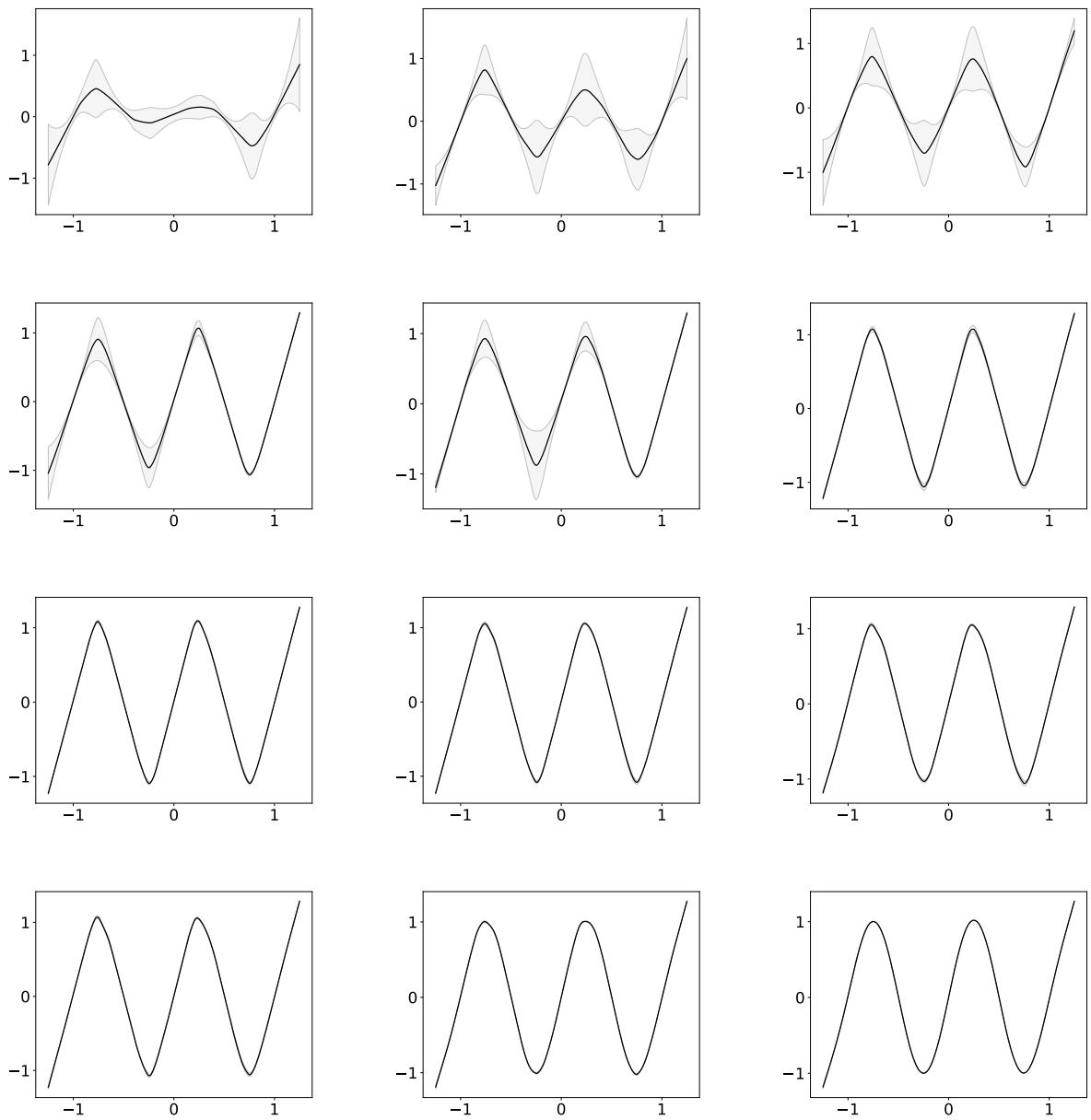


Figure B.13. Visualization of the mean prediction and variance of the different width neural networks. Widths in increasing order from left to right and top to bottom: 5, 10, 15, 17, 20, 22, 25, 35, 75, 100, 1000, 10000.

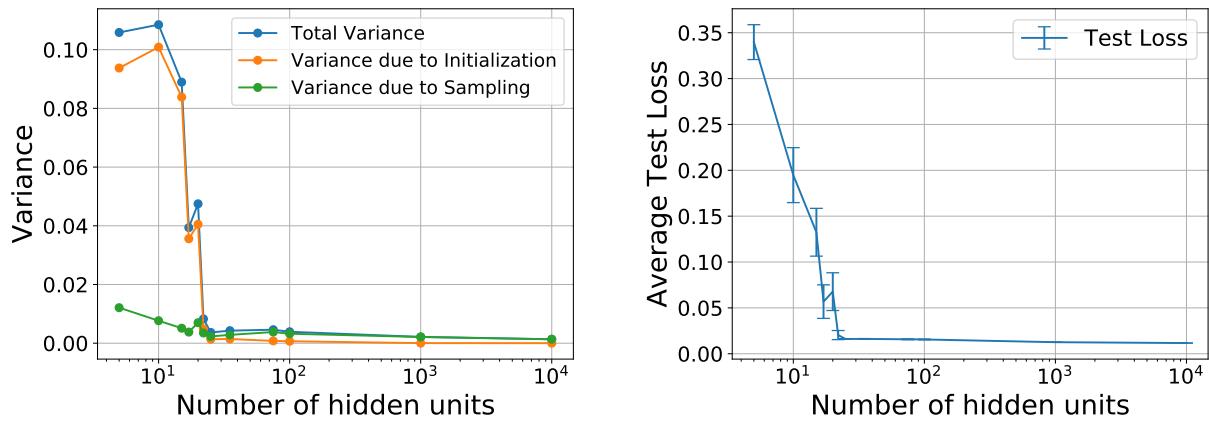
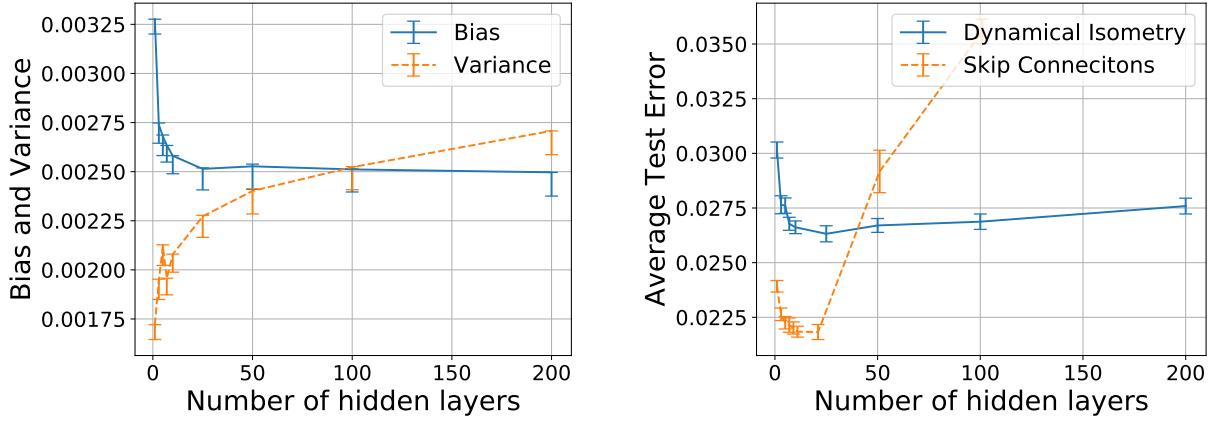


Figure B.14. We observe the same trends of decomposed variance (left) and test error (right) in the sinusoid regression setting.

Appendix C

Depth and variance

C.1. Main graphs



(a) Bias and variance trends with depth, using dynamical isometry (b) Test error trends, using dynamical isometry vs. skip connections

Figure C.1. We can see that, when using dynamical isometry, bias decreases with depth and variance slowly increases with depth (left). This increase in variance is so small that it only translates to a an increase in test error of about 0.1% for depth 25 to depth 200 (right).

C.2. Discussion on need for careful experimental design

Depth is an important component of deep learning. We study its effect on bias and variance by fixing width and varying depth. However, there are pathological problems associated with training very deep networks such as vanishing/exploding gradient (Hochreiter, 1991; Bengio et al., 1994; Glorot & Bengio, 2010), signal not being able to propagate through the network (Schoenholz et al., 2017), and gradients resembling white noise (Balduzzi et al., 2017). He et al. (2016) pointed out that very deep networks experience high test set error and argued it was due to high training set loss. However, while skip connections (He et al., 2016), better initialization (Glorot & Bengio, 2010), and batch normalization (Ioffe &

Szegedy, 2015) have largely served to facilitate low training loss in very deep networks, the problem of high *test set* error still remains.

The current best practices for achieving low test error in very deep networks arose out of trying to solve the above problems in training. An initial step was to ensure the mean squared singular value of the input-output Jacobian, at initialization, is close to 1 (Glorot & Bengio, 2010). More recently, there has been work on a stronger condition known as *dynamical isometry*, where *all* singular values remain close to 1 (Saxe et al., 2014; Pennington et al., 2017). Pennington et al. (2017) also empirically found that dynamical isometry helped achieve low test set error. Furthermore, Xiao et al. (2018, Figure 1) found evidence that test set performance did not degrade with depth when they lifted dynamical isometry to CNNs. This why we settled on dynamical isometry as the best known practice to control for as many confounding factors as possible.

We first ran experiments with vanilla full connected networks (Figure C.2). These have clear training issues where networks of depth more than 20 take very long to train to the target training loss of 5e-5. The bias curve is not even monotonically decreasing. Clearly, there are important confounding factors not controlled for in this simple setting. Still, note that variance increases roughly linearly with depth.

We then study fully connected networks with skip connections between every 2 layers (Figure C.3). While this allows us to train deeper networks than without skip connections, many of the same issues persist (e.g. bias still not monotonically decreasing). The bias, variance, and test error curves are all checkmark-shaped.

C.3. Vanilla fully connected depth experiments

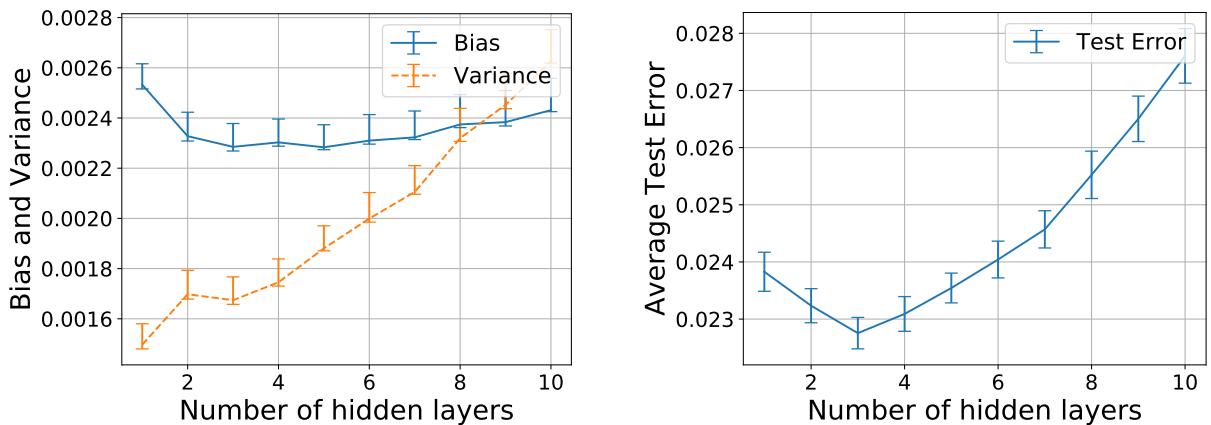


Figure C.2. Test error quickly degrades in fairly shallow fully connected networks, and bias does not even monotonically decrease with depth. However, this is the first indication that variance might *increase* with depth. All networks have training error 0 and are trained to the same training loss of 5e-5.

C.4. Skip connections depth experiments

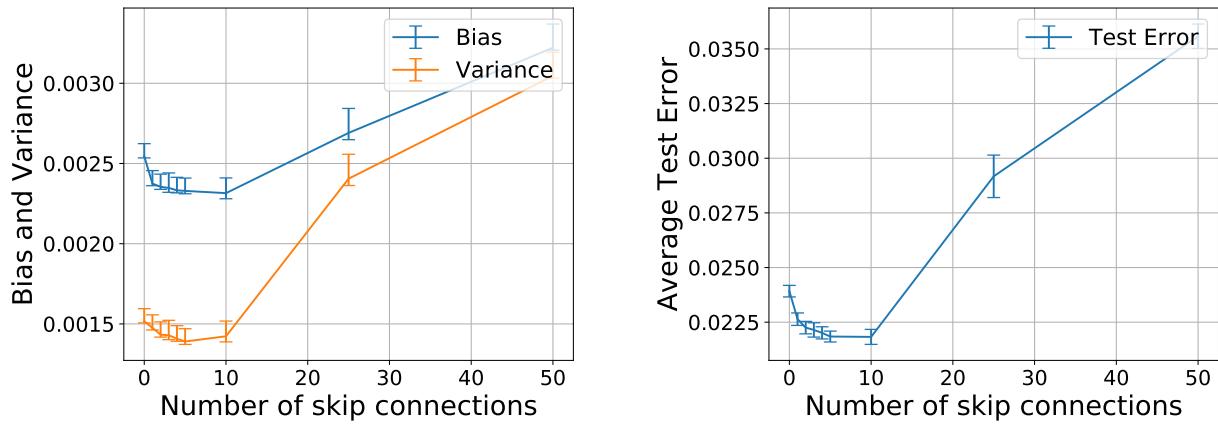


Figure C.3. While the addition of skip connections (between every other layer) might push the bottom of the U curve in test error out to 10 skip connections (21 layers), which is further than where the bottom was observed without skip connections (3 layers), test error still degrades noticeably in greater depths. Additionally, bias still does not even monotonically decrease with depth. While skip connections appear to have helped control for the factors we want to control, they were not completely satisfying. All networks have training error 0 and are trained to the same training loss of 5e-5.

C.5. Dynamical isometry depth experiments

The figures in this section are included in the main paper, but they are included here for comparison to the above and for completeness.

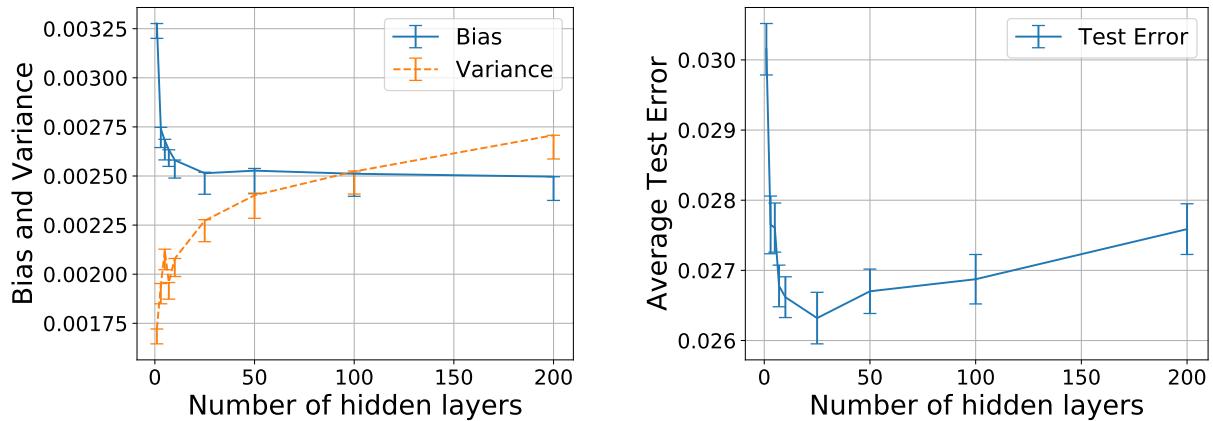


Figure C.4. Additionally, dynamical isometry seems to cause bias to decrease monotonically with depth. While skip connections appear to have helped control for the factors we want to control, they were not completely satisfying. All networks have training error 0 and are trained to the same training loss of 5e-5.

Appendix D

Some Proofs

D.1. Proof of Classic Result for Variance of Linear Model

Here, we reproduce the classic result that variance grows with the number of parameters in a linear model. This result can be found in [Hastie et al. \(2009\)](#)'s book, and a similar proof can be found in [Gonzalez \(2016\)](#)'s lecture slides.

PROOF. For a fixed x , we have $h(x) = x^T \hat{\theta}$. Taking $\hat{\theta} = \Sigma^{-1} X^T Y$ to be the gradient descent solution, and using $Y = X\theta + \epsilon$, we obtain:

$$h(x) = x^T \Sigma^{-1} X^T (X\theta + \epsilon) = x^T \theta + x^T \Sigma^{-1} X^T \epsilon$$

Hence $\mathbb{E}_\epsilon[h(x)] = x^T \theta$, and the variance is,

$$\begin{aligned} \text{Var}_\epsilon(h(x)) &= \mathbb{E}_\epsilon[(h(x) - \mathbb{E}_\epsilon[h(x)])^2] \\ &= \mathbb{E}_\epsilon[(x^T \theta + x^T \Sigma^{-1} X^T \epsilon - x^T \theta)^2] \\ &= \mathbb{E}_\epsilon[(x^T \Sigma^{-1} X^T \epsilon)^2] \\ &= \mathbb{E}_\epsilon[(x^T \Sigma^{-1} X^T \epsilon)(x^T \Sigma^{-1} X^T \epsilon)^T] \\ &= \mathbb{E}_\epsilon[x^T \Sigma^{-1} X^T \epsilon \epsilon^T (x^T \Sigma^{-1} X^T)^T] \\ &= \sigma_\epsilon^2 x^T \Sigma^{-1} \Sigma \Sigma^{-1} x \\ &= \sigma_\epsilon^2 x^T \Sigma^{-1} \Sigma \Sigma^{-1} x \\ &= \sigma_\epsilon^2 x^T \Sigma^{-1} x \\ &= \sigma_\epsilon^2 \text{Tr}(x^T \Sigma^{-1} x) \\ &= \sigma_\epsilon^2 \text{Tr}(x x^T \Sigma^{-1}) \end{aligned}$$

Taking the expected value over the empirical distribution, \hat{p} , of the sample, we find an explicit increasing dependence on N :

$$\begin{aligned}
\mathbb{E}_{x \sim \hat{p}}[\text{Var}_\epsilon(h(x))] &= \mathbb{E}_{x \sim \hat{p}}[\sigma_\epsilon^2 \text{Tr}(xx^T \Sigma^{-1})] \\
&= \sigma_\epsilon^2 \text{Tr}(\mathbb{E}_{x \sim \hat{p}}[xx^T] \Sigma^{-1}) \\
&= \sigma_\epsilon^2 \text{Tr}\left(\frac{1}{m} \Sigma \Sigma^{-1}\right) \\
&= \sigma_\epsilon^2 \frac{1}{m} \text{Tr}(I_N) \\
&= \sigma_\epsilon^2 \frac{N}{m}
\end{aligned}$$

□

D.2. Proof of Result for Variance of Over-parameterized Linear Models

Here, we produce a variation on what was done in Appendix D.1 to show that variance does not grow with the number of parameters in over-parameterized linear models. Recall that we are considering the setting where $N > m$, where N is the number of parameters and m is the number of training examples.

PROOF. By the law of total variance,

$$\text{Var}(h(x)) = \mathbb{E}_\epsilon \text{Var}_{\theta_0}(h(x)) + \text{Var}_\epsilon(\mathbb{E}_{\theta_0}[h(x)])$$

Here have $h(x) = x^T \hat{\theta}$, where $\hat{\theta}$ the gradient descent solution $\hat{\theta} = P_\perp(\theta_0) + \Sigma^+ X^T Y$, and $\theta_0 \sim \mathcal{N}(0, \frac{1}{N} I)$. Then,

$$\begin{aligned}
\text{Var}_{\theta_0}(h(x)) &= \mathbb{E}_{\theta_0}[(h(x) - \mathbb{E}_{\theta_0}[h(x)])^2] \\
&= \mathbb{E}_{\theta_0}[x^T (P_\perp(\theta_0) - \mathbb{E}_{\theta_0}[P_\perp(\theta_0)])^2] \\
&= \text{Var}_{\theta_0}(x^T P_\perp(\theta_0)) \\
&= \text{Var}_{\theta_0}(P_\perp(x)^T P_\perp(\theta_0)) \\
&= \frac{1}{N} \|P_\perp(x)\|^2
\end{aligned}$$

Since $\mathbb{E}_{\theta_0}(h(x)) = x^T \Sigma^+ X^T Y$, the calculation of $\text{Var}_\epsilon(\mathbb{E}_{\theta_0} h(x))$ is similar as in D.1, where Σ^{-1} is replaced by Σ^+ . Thus,

$$\text{Var}_\epsilon(\mathbb{E}_{\theta_0} h(x)) = \sigma_\epsilon^2 \text{Tr}(xx^T \Sigma^+)$$

Taking the expected value over the empirical distribution, \hat{p} , of the sample, we find an explicit dependence on $r = \text{rank}(X)$, not N :

$$\begin{aligned}\mathbb{E}_{x \sim \hat{p}}[\text{Var}(h(x))] &= 0 + \mathbb{E}_{x \sim \hat{p}}[\sigma_\epsilon^2 \text{Tr}(xx^T \Sigma^+)] \\ &= \sigma_\epsilon^2 \text{Tr}(\mathbb{E}_{x \sim \hat{p}}[xx^T] \Sigma^+) \\ &= \sigma_\epsilon^2 \text{Tr}\left(\frac{1}{m} \Sigma \Sigma^+\right) \\ &= \sigma_\epsilon^2 \frac{1}{m} \text{Tr}(I_r^+) \\ &= \sigma_\epsilon^2 \frac{r}{m}\end{aligned}$$

where I_r^+ denotes the diagonal matrix with 1 for the first r diagonal elements and 0 for the remaining $N - r$ elements. \square

D.3. Proof of Theorem 5.5.2

First we state some known concentration results (Ledoux, 2001) that we will use in the proof.

Lemma D.3.1 (Levy). *Let $h : S_R^n \rightarrow \mathbb{R}$ be a function on the n -dimensional Euclidean sphere of radius R , with Lipschitz constant L ; and $\theta \in S_R^n$ chosen uniformly at random for the normalized measure. Then*

$$\mathbb{P}(|h(\theta) - \mathbb{E}[h]| > \epsilon) \leq 2 \exp\left(-C \frac{n\epsilon^2}{L^2 R^2}\right) \quad (\text{D.3.1})$$

for some universal constant $C > 0$.

Uniform measures on high dimensional spheres approximate Gaussian distributions (Ledoux, 2001). Using this, Levy's lemma yields an analogous concentration inequality for functions of Gaussian variables:

Lemma D.3.2 (Gaussian concentration). *Let $h : \mathbb{R}^n \rightarrow \mathbb{R}$ be a function on the Euclidean space \mathbb{R}^n , with Lipschitz constant L ; and $\theta \sim \mathcal{N}(0, \sigma^2 \mathbb{I}_n)$ sampled from an isotropic n -dimensional Gaussian. Then:*

$$\mathbb{P}(|h(\theta) - \mathbb{E}[h]| > \epsilon) \leq 2 \exp\left(-C \frac{\epsilon^2}{L^2 \sigma^2}\right) \quad (\text{D.3.2})$$

for some universal constant $C > 0$.

Note that in the Gaussian case, the bound is dimension free.

In turn, concentration inequalities give variance bounds for functions of random variables.

Corollary D.3.3. *Let h be a function satisfying the conditions of Theorem D.3.2, and $\text{Var}(h) = \mathbb{E}[(h - \mathbb{E}[h])^2]$. Then*

$$\text{Var}(h) \leq \frac{2L^2 \sigma^2}{C} \quad (\text{D.3.3})$$

PROOF. Let $g = h - \mathbb{E}[h]$. Then $\text{Var}(h) = \text{Var}(g)$ and

$$\text{Var}(g) = \mathbb{E}[|g|^2] = 2\mathbb{E} \int_0^{|g|} t dt = 2\mathbb{E} \int_0^\infty t \mathbb{1}_{|g|>t} dt \quad (\text{D.3.4})$$

Now swapping expectation and integral (by Fubini theorem), and by using the identity $\mathbb{E}\mathbb{1}_{|g|>t} = \mathbb{P}(|g| > t)$, we obtain

$$\begin{aligned} \text{Var}(g) &= 2 \int_0^\infty t \mathbb{P}_R(|g| > t) dt \\ &\leq 2 \int_0^\infty 2t \exp\left(-C \frac{t^2}{L^2 \sigma^2}\right) dt \\ &= 2 \left[-\frac{L^2 \sigma^2}{C} \exp\left(-C \frac{t^2}{L^2 \sigma^2}\right) \right]_0^\infty = \frac{2L^2 \sigma^2}{C} \end{aligned}$$

□

We are now ready to prove Theorem 5.5.2. We first recall our assumptions:

Assumption 1. *The optimization of the loss function is invariant with respect to $\theta_{\mathcal{M}^\perp}$.*

Assumption 2. *Along \mathcal{M} , optimization yields solutions independently of the initialization θ_0 .*

We add the following assumptions.

Assumption 3. *The prediction $h_\theta(x)$ is L -Lipschitz with respect to $\theta_{\mathcal{M}^\perp}$.*

Assumption 4. *The network parameters are initialized as*

$$\theta_0 \sim \mathcal{N}(0, \frac{1}{N} \cdot I_{N \times N}). \quad (\text{D.3.5})$$

We first prove that the Gaussian concentration theorem translates into concentration of predictions in the setting of Section 5.5.2.1.

Theorem D.3.4 (Concentration of predictions). *Consider the setting of Section 5.5.2 and Assumptions 1 and 4. Let θ denote the parameters at the end of the learning process. Then, for a fixed data set, S we get concentration of the prediction, under initialization randomness,*

$$\mathbb{P}(|h_\theta(x) - \mathbb{E}[h_\theta(x)]| > \epsilon) \leq 2 \exp\left(-C \frac{N\epsilon^2}{L^2}\right) \quad (\text{D.3.6})$$

for some universal constant $C > 0$.

PROOF. In our setting, the parameters at the end of learning can be expressed as

$$\theta = \theta_{\mathcal{M}}^* + \theta_{\mathcal{M}^\perp} \quad (\text{D.3.7})$$

where $\theta_{\mathcal{M}}^*$ is independent of the initialization θ_0 . To simplify notation, we will assume that, at least locally around $\theta_{\mathcal{M}}^*$, \mathcal{M} is spanned by the first $d(N)$ standard basis vectors, and \mathcal{M}^\perp by the remaining $N - d(N)$. This will allow us, from now on, to use the same variable names for $\theta_{\mathcal{M}}$ and $\theta_{\mathcal{M}^\perp}$ to denote their lower-dimensional representations of dimension $d(N)$ and

$N - d(N)$ respectively. More generally, we can assume that there is a mapping from $\theta_{\mathcal{M}}$ and $\theta_{\mathcal{M}^\perp}$ to those lower-dimensional representations.

From Assumptions 1 and 4 we get

$$\theta_{\mathcal{M}^\perp} \sim \mathcal{N}\left(0, \frac{1}{N} I_{(N-d(N)) \times (N-d(N))}\right). \quad (\text{D.3.8})$$

Let $g(\theta_{\mathcal{M}^\perp}) \triangleq h_{\theta_{\mathcal{M}}^* + \theta_{\mathcal{M}^\perp}}(x)$. By Assumption 3, $g(\cdot)$ is L -Lipschitz. Then, by the Gaussian concentration theorem we get,

$$\mathbb{P}(|g(\theta_{\mathcal{M}^\perp}) - \mathbb{E}[g(\theta_{\mathcal{M}^\perp})]| > \epsilon) \leq 2 \exp\left(-C \frac{N\epsilon^2}{L^2}\right). \quad (\text{D.3.9})$$

□

The result of Theorem 5.5.2 immediately follows from Theorem D.3.4 and Corollary D.3.3, with $\sigma^2 = 1/N$:

$$\text{Var}_{\theta_0}(h_\theta(x)) \leq C \frac{2L^2}{N} \quad (\text{D.3.10})$$

Provided the Lipschitz constant L of the prediction grows more slowly than the square of dimension, $L = o(\sqrt{N})$, we conclude that the variance vanishes to zero as N grows.

D.4. Bound on classification error in terms of regression error

In this section we give a bound on classification risk $\mathcal{R}_{\text{classif}}$ in terms of the regression risk \mathcal{R}_{reg} .

Notation. Our classifier defines a map $h : \mathcal{X} \rightarrow \mathbb{R}^k$, which outputs probability vectors $h(x) \in \mathbb{R}^k$, with $\sum_{y=1}^k h(x)_y = 1$. The classification loss is defined by

$$\begin{aligned} L(h) &= \text{Prob}_{x,y}\{h(x)_y < \max_{y'} h(x)_{y'}\} \\ &= \mathbb{E}_{(x,y)} I(h(x)_y < \max_{y'} h(x)_{y'}) \end{aligned} \quad (\text{D.4.1})$$

where $I(a) = 1$ if predicate a is true and 0 otherwise. Given trained predictors h_S indexed by training dataset S , the classification and regression risks are given by,

$$\mathcal{R}_{\text{classif}} = \mathbb{E}_S L(h_S), \quad \mathcal{R}_{\text{reg}} = \mathbb{E}_S \mathbb{E}_{(x,y)} \|h_S(x) - Y\|_2^2 \quad (\text{D.4.2})$$

where Y denotes the one-hot vector representation of the class y .

Proposition D.4.1. *The classification risk is bounded by four times the regression risk, $\mathcal{R}_{\text{classif}} \leq 4\mathcal{R}_{\text{reg}}$.*

PROOF. First note that, if $h(x) \in \mathbb{R}^k$ is a probability vector, then

$$h(x)_y < \max_{y'} h(x)_{y'} \implies h(x)_y < \frac{1}{2}$$

By taking the expectation over x, y , we obtain the inequality $L(h) \leq \tilde{L}(h)$ where

$$\tilde{L}(h) = \text{Prob}_{x,y}\{h(x)_y < \frac{1}{2}\} \quad (\text{D.4.3})$$

We then have,

$$\begin{aligned} \mathcal{R}_{\text{classif}} &:= \mathbb{E}_S L(h_S) \leq \mathbb{E}_S \tilde{L}(h_S) \\ &= \text{Prob}_{S;x,y}\{h_S(x)_y < \frac{1}{2}\} \\ &= \text{Prob}_{S;x,y}\{|h_S(x)_y - Y_y| > \frac{1}{2}\} \\ &\leq \text{Prob}_{S;x,y}\{\|h_S(x) - Y\|_2 > \frac{1}{2}\} \\ &= \text{Prob}_{S;x,y}\{\|h_S(x) - Y\|_2^2 > \frac{1}{4}\} \leq 4\mathcal{R}_{\text{reg}} \end{aligned}$$

where the last inequality follows from Markov's inequality.

□

Appendix E

Common intuitions from impactful works

“Neural Networks and the Bias/Variance Dilemma” from ([Geman et al., 1992](#)): “How big a network should we employ? A small network, with say one hidden unit, is likely to be biased, since the repertoire of available functions spanned by $f(x; w)$ over allowable weights will in this case be quite limited. If the true regression is poorly approximated within this class, there will necessarily be a substantial bias. On the other hand, if we overparameterize, via a large number of hidden units and associated weights, then the bias will be reduced (indeed, with enough weights and hidden units, the network will interpolate the data), but there is then the danger of a significant variance contribution to the mean-squared error. (This may actually be mitigated by incomplete convergence of the minimization algorithm, as we shall see in Section 3.5.5.)”

“An Overview of Statistical Learning Theory” from ([Vapnik, 1999](#)): “To avoid over fitting (to get a small confidence interval) one has to construct networks with small VC-dimension.”

“Stability and Generalization” from [Bousquet & Elisseeff \(2002\)](#): “It has long been known that when trying to estimate an unknown function from data, one needs to find a tradeoff between bias and variance. Indeed, on one hand, it is natural to use the largest model in order to be able to approximate any function, while on the other hand, if the model is too large, then the estimation of the best function in the model will be harder given a restricted amount of data.” Footnote: “We deliberately do not provide a precise definition of bias and variance and resort to common intuition about these notions.”

Pattern Recognition and Machine Learning from [Bishop \(2006\)](#): “Our goal is to minimize the expected loss, which we have decomposed into the sum of a (squared) bias, a variance, and a constant noise term. As we shall see, there is a trade-off between bias and variance, with very flexible models having low bias and high variance, and relatively rigid models having high bias and low variance.”

“Understanding the Bias-Variance Tradeoff” from [Fortmann-Roe \(2012\)](#): “At its root, dealing with bias and variance is really about dealing with over- and under-fitting. Bias is reduced and variance is increased in relation to model complexity. As more and more

parameters are added to a model, the complexity of the model rises and variance becomes our primary concern while bias steadily falls. For example, as more polynomial terms are added to a linear regression, the greater the resulting model's complexity will be.”

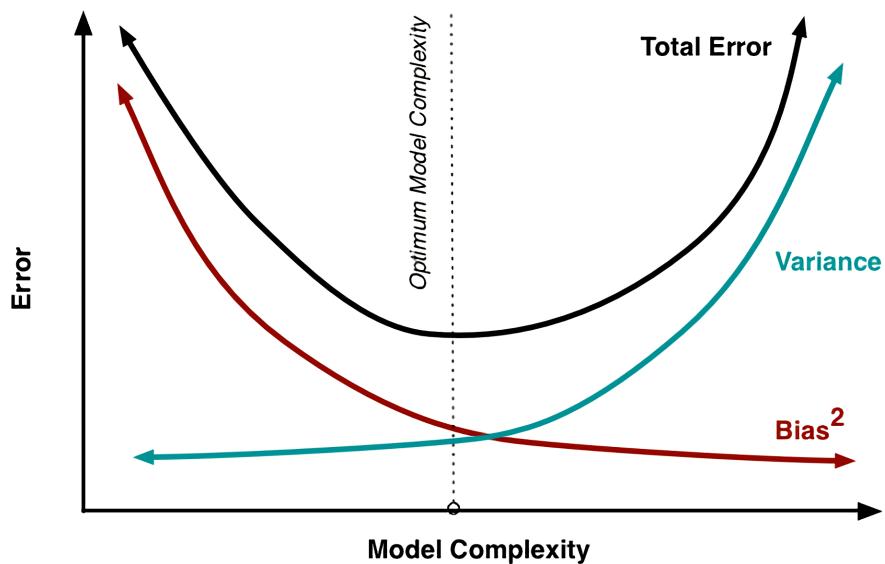


Figure E.1. Illustration of common intuition for bias-variance tradeoff (Fortmann-Roe, 2012)