Rethinking Bias-Variance Trade-off for Generalization of Neural Networks

Zitong Yang 1, Yaodong Yu 1, Chong You 1, Jacob Steinhardt 1,2, Yi Ma 1

Abstract
The classical bias-variance trade-off predicts that bias decreases and variance increases with model complexity, leading to a U-shaped risk curve. Recent work calls this into question for neural networks and other over-parameterized models, for which it is often observed that larger models generalize better. We provide a simple explanation for this by measuring the bias and variance of neural networks: while the bias is monotonically decreasing as in the classical theory, the variance is unimodal or bell-shaped: it increases then decreases with the width of the network. We vary the network architecture, loss function, and choice of dataset and confirm that variance unimodality occurs robustly for all models we considered. The risk curve is the sum of the bias and variance curves and displays different qualitative shapes depending on the relative scale of bias and variance, with the double descent curve observed in recent literature as a special case. We corroborate these empirical results with a theoretical analysis of two-layer linear networks with random first layer. Finally, evaluation on out-of-distribution data shows that most of the drop in accuracy comes from increased bias while variance increases by a relatively small amount. Moreover, we find that deeper models decrease bias and increase variance for both in-distribution and out-of-distribution data.

1. Introduction
Bias-variance trade-off is a fundamental principle for understanding the generalization of predictive learning models (Hastie et al., 2001). The bias is an error term that stems from a mismatch between the model class and the underlying data distribution, and is typically monotonically non-increasing as a function of the complexity of the model. The variance measures sensitivity to fluctuations in the training set and is often attributed to a large number of model parameters. Classical wisdom predicts that model variance increases and bias decreases monotonically with model complexity (Geman et al., 1992). Under this perspective, we should seek a model that has neither too little nor too much capacity and achieves the best trade-off between bias and variance.

In contrast, modern practice for neural networks repeatedly demonstrates the benefit of increasing the number of neurons (Krizhevsky et al., 2012; Simonyan & Zisserman, 2015; Zhang et al., 2017), even up to the point of saturating available memory. The inconsistency between classical theory and modern practices suggests that some arguments in the classical theory can not be applied to modern neural networks.

Geman et al. (1992) first studied the bias and variance of the neural networks and give experimental evidence that the variance is indeed increasing as the width of the neural network increases. Since Geman et al. (1992), Neal et al. (2019) first experimentally measured the variance of modern neural network architectures and shown that the variance can actually be decreasing as the width increases to a highly overparameterized regime. Recently, Belkin et al. (2019a; 2018; 2019b) directly studied the risk of modern machine learning models and proposed a double descent risk curve, which has also been analytically characterized for certain regression and classification models (Mei & Montanari, 2019; Hastie et al., 2019; Spigler et al., 2019; Deng et al., 2019; Advani & Saxe, 2017; Bartlett et al., 2020; Chatterji & Long, 2020). However, there exists two mysteries around the double descent risk curve. First, the double descent phenomenon can not be robustly observed (Nakkiran et al., 2019; Ba et al., 2020). In particular, to observe it in modern neural network architectures, we sometimes have to artificially inject label noise (Nakkiran et al., 2019). Second, there lacks an explanation for why the double descent risk curve should occur. In this work, we offer an simple explanation for these two mysteries by proposing an unexpected unimodal variance curve.
Specifically, we measure the bias and variance of modern deep neural networks trained on commonly used computer vision datasets. Our main finding is that while the bias is monotonically decreasing with network width as in the classical theory, the variance curve is unimodal or bell-shaped: it first increases and then decreases (see Figure 2). Therefore, the unimodal variance is consistent with the finding of Neal et al. (2019), who observed that the variance eventually decreases in the overparameterized regime. In particular, the unimodal variance curve can also be observed in Neal et al. (2019, Figure 1, 2, 3). However, Neal et al. (2019) did not point out the characteristic shape of the variance or connect it to double descent. More importantly, we demonstrate that the unimodal variance phenomenon can be robustly observed for varying network architecture and dataset. Moreover, by using a generalized bias-variance decomposition for Bregman divergences (Pfau, 2013), we verify that it occurs for both squared loss and cross-entropy loss.

This unimodal variance phenomenon initially appears to contradict recent theoretical work suggesting that both bias and variance are non-monotonic and exhibit a peak in some regimes (Mei & Montanari, 2019; Hastie et al., 2019). The difference is that this previous work considered the fixed-design bias and variance, while we measure the random-design bias and variance (we describe the differences in detail in §2.1). Prior to our work, Nakkiran (2019) also considered the variance of linear regression in the random-design setting, and Rosset & Tibshirani (2017) discussed additional ways to decompose risk into the bias and the variance term.

A key finding of our work is that the complex behavior of the risk curve arises due to the simple but non-classical variance unimodality phenomenon. Indeed, since the expected risk (test loss) is the sum of bias and variance, monotonic bias and unimodal variance can lead to three characteristic behaviors, illustrated in Figure 1, depending on the relative size of the bias and variance. If the bias completely dominates, we obtain monotonically decreasing risk curve (see Figure 1(a)). Meanwhile, if the variance dominates, we obtain a bell-shaped risk curve that first increases then decreases (see Figure 1(c)). The most complex behavior is if bias and variance dominate in different regimes, leading to the double-descent risk curve in Figure 1(b). All three behaviors are well-aligned with the empirical observation in deep learning that larger models typically perform better. The most common behavior in our experiments is the first case (monotonically decreasing risk curve) as bias is typically larger than variance. We can observe the double-descent risk curve when label noise is added to the training set (see §3.3), and can observe the unimodal risk curve when we use the generalized bias-variance decomposition for cross-entropy loss (see §3.2).

**Further Implications.** The investigations described above characterize bias and variance as a function of network width, but we can explore the dependence on other quantities as well, such as model depth (§4.2). Indeed, we find that deeper models tend to have lower bias but higher variance. Since bias is larger at current model sizes, this confirms the prevailing wisdom that we should generally use deeper models when possible. On the other hand, it suggests that this process may have a limit—eventually very deep models may have low bias but high variance such that increasing the depth further harms performance.

We also investigate the commonly observed drop in accuracy for models evaluated on out-of-distribution data, and attribute it primarily to increased bias. Combined with the previous observation, this suggests that increasing model depth may help combat the drop in out-of-distribution accuracy, which is supported by experimental findings in Hendrycks & Dietterich (2019).

**Theoretical Analysis of A Two-Layer Neural Network.** Finally, we conduct a theoretical study of a two-layer linear network with a random Gaussian first layer. While this model is much simpler than those used in practice, we nevertheless observe the same characteristic behaviors for the bias and variance. In particular, by working in the asymptotic setting where the input data dimension, amount of training data, and network width go to infinity with fixed ratios, we show that the bias is monotonically decreasing while the...
variance curve is unimodal. Our analysis also characterizes the location of the variance peak as the point where the number of hidden neurons is approximately half of the dimension of the input data.

2. Preliminaries

In this section we present the bias-variance decomposition for squared loss. We also present a generalized bias-variance decomposition for cross-entropy loss in §2.2. The task is to learn a function \( f : \mathbb{R}^d \rightarrow \mathbb{R} \), based on i.i.d. training samples \( T = \{(x_i, y_i)\}_{i=1}^n \) drawn from a joint distribution \( P \) on \( \mathbb{R}^d \times \mathbb{R} \), such that the mean squared error \( \mathbb{E}_{x,y} (\| y - f(x, T) \|_2^2) \) is minimal, where \( (x, y) \sim P \). Here we denote the learned function by \( f(x; T) \) to make the dependence on the training samples clear.

Note that the learned predictor \( f(x; T) \) is a random quantity depending on \( T \). We can assess its performance in two different ways. The first way, random-design, takes the expectation over \( T \) such that we consider the expected error \( \mathbb{E}_T (\| y - f(x, T) \|_2^2) \). The second way, fixed-design, holds the training covariates \( \{x_i\}_{i=1}^n \) fixed and only takes expectation over \( \{y_i\}_{i=1}^n \), i.e., \( \mathbb{E}_x \mathbb{E}_y (\| y - f(x, T) \|_2^2 | \{x_i\}_{i=1}^n) \). The choice of random/fixed-design leads to different bias-variance decompositions. Throughout the paper, we focus on random-design, as opposed to fixed-design studied in Mei & Montanari (2019); Hastie et al. (2019); Ba et al. (2020).

2.1. Bias Variance Decomposition

**Random Design.** In the random-design setting, decomposing the quantity \( \mathbb{E}_T (\| y - f(x, T) \|_2^2) \) gives the usual bias-variance trade-off from machine learning, e.g. Geman et al. (1992); Hastie et al. (2001).

\[
\begin{align*}
\mathbb{E}_{x,y} \mathbb{E}_T (\| y - f(x, T) \|_2^2) &= \mathbb{E}_{x,y} \mathbb{E}_T (\| y - f(x, T) \|_2^2) + \mathbb{E}_{x} \mathbb{E}_T (\| f(x, T) - \bar{f}(x) \|_2^2).
\end{align*}
\]

where \( \bar{f}(x) = \mathbb{E}_T f(x, T) \). Here \( \mathbb{E}_T (\| y - f(x, T) \|_2^2) \) measures the average prediction error over different realizations of the training sample. In addition to take the expectation \( \mathbb{E}_T \), we also average over \( \mathbb{E}_{x,y} \), as discussed in Bishop (2006, §3.2). For future reference, we define

\[
\begin{align*}
\text{Bias}^2 &= \mathbb{E}_{x,y} (\| y - \bar{f}(x) \|_2^2), \\
\text{Variance} &= \mathbb{E}_{x} \mathbb{E}_T (\| f(x, T) - \bar{f}(x) \|_2^2).
\end{align*}
\]

In §2.2, we present our estimator for bias and variance in equation (1) and (2).

**Fixed Design.** In fixed-design setting, the covariates \( \{x_i\}_{i=1}^n \) are held be fixed, and the only randomness in the training set \( T \) comes from \( y_i \sim P(Y \mid X = x_i) \). As presented in Mei & Montanari (2019); Hastie et al. (2019); Ba et al. (2020), a more natural way to present the fixed-design assumption is to hold \( \{x_i\}_{i=1}^n \) to be fixed and let \( y_i = f_0(x) + \epsilon_i \) for \( i = 1, \ldots, n \), where \( f_0(x) \) is a ground-truth function and \( \epsilon_i \) are random noises. Under this assumption, the randomness in \( T \) all comes from the random noise \( \epsilon_i \). To make this clear, we write \( T \) as \( T_{\epsilon} \). Then, we obtain the fixed-design bias-variance decomposition

\[
\begin{align*}
\mathbb{E}_{\epsilon_i} (\| (y - f(x, T_{\epsilon}) \|_2^2) &= \mathbb{E}_{\epsilon_i} (\| (y - \bar{f}(x) \|_2^2) + \mathbb{E}_{\epsilon_i} (\| (f(x, T_{\epsilon}) - \bar{f}(x) \|_2^2),
\end{align*}
\]

where \( \bar{f}(x) = \mathbb{E}_{\epsilon_i} f(x, T_{\epsilon}) \). In most practical settings, the expectation \( \mathbb{E}_{\epsilon_i} f(x, T_{\epsilon}) \) cannot be estimated from training samples \( T = \{ (x_i, y_i) \}_{i=1}^n \), because we do not have access to independent copies of \( f(x_i) + \epsilon_i \). In comparison to the random-design setting, the fixed-design setting tends to have larger bias and smaller variance, since more “randomness” is introduced into the variance term.

2.2. Estimating Bias and Variance

In this section, we present the estimator we use to estimate the bias and variance as defined in equation (1) and (2). The high level idea is to approximate the expectation \( \mathbb{E}_T \) by computing the sample average using multiple training sets \( T_1, \ldots, T_N \). When evaluating the expectation \( \mathbb{E}_T \), there is a trade-off between having larger training sets (\( n \)) within each training set and having larger number of splits (\( N \)), since \( n \times N = \) total number of training samples.

**Mean Squared Error (MSE).** To estimate bias and variance in equation (1) and (2), we introduce an unbiased estimator for variance, and obtain bias by subtracting the variance from the risk. Let \( T = T_1 \cup \ldots \cup T_N \) be a random disjoint split of training samples. In our experiment, we mainly take \( N = 2 \) (for CIFAR10 each \( T_i \) has 25k samples). To estimate the variance, we use the unbiased estimator

\[
\text{Var}(x, T) = \frac{1}{N - 1} \sum_{j=1}^{N} \| f(x, T_j) - \frac{1}{N} \sum_{j=1}^{N} f(x, T_j) \|_2^2,
\]

where \( \bar{f} \) depends on the test point \( x \) and on the random training set \( T \). While \( \text{Var} \) is unbiased, its variance can be reduced by using multiple random splits to obtain estimators \( \text{Var}_1, \ldots, \text{Var}_k \) and taking their average. This reduces the variance of the variance estimator since:

\[
\text{Var}_T \left( \frac{1}{k} \sum_{i=1}^{k} \text{Var}_i \right) \leq \text{Var}_T (\text{Var}_1),
\]

where the \( \{ \text{Var}_i \}_{i=1}^{k} \) are identically distributed but not independent, and we used the Cauchy-Schwarz inequality.
Cross-Entropy Loss (CE). In addition to the classical bias-variance decomposition for MSE loss, we also consider a generalized bias-variance decomposition for cross-entropy loss. Let \( \pi(x, T) \in \mathbb{R}^c \) be the output of the neural network (a probability distribution over the class labels). \( \pi(x, T) \) is a random variable since the training set \( T \) is random. Let \( \pi_0(x) \in \mathbb{R}^c \) be the one-hot encoding of the ground-truth label. Then, omitting the dependence of \( \pi \) and \( \pi_0 \) on \( x \) and \( T \), the cross entropy loss

\[
H(\pi_0, \pi) = \sum_{l=1}^{c} \pi_0[l] \log(\pi[l])
\]

can be decomposed as

\[
E_T [H(\pi_0, \pi)] = \underbrace{D_{KL}(\pi_0 || \bar{\pi})}_{\text{Bias}} + \underbrace{D_{KL}(\bar{\pi} || \pi)}_{\text{Variance}},
\]

where \( \pi[l] \) is the \( l \)-th element of \( \pi \), and \( \bar{\pi} \) is the average of log-probability after normalization, i.e.,

\[
\bar{\pi}[l] \propto \exp[\mathbb{E}_T \log(\pi[l])] \quad \text{for} \quad l = 1, \ldots, c.
\]

This decomposition is a special case of the general decomposition for Bregman divergence discussed in Pfau (2013).

We apply Algorithm 1 to estimate the generalized variance in (3). Here we could not obtain an unbiased estimator, but the estimate is better if we take more random splits (larger \( k \)). In practice, we choose \( k \) to be large enough so that the estimated variance stabilizes when we further increase \( k \) (see §3.4). Similar to the case of squared loss, we estimate the bias by subtracting the variance from the risk.

### 3. Measuring Bias and Variance for Neural Networks

In this section, we study the bias and variance (equations (1) and (2)) of deep neural networks. While the bias is monotonically decreasing as folk wisdom would predict, the variance is unimodal (first increases to a peak and then decreases). We conduct extensive experiments to verify that this phenomenon appears robustly across architectures, datasets, optimizer, and loss function. Our code can be found at https://github.com/yaodongyu/Rethink-BiasVariance-Tradeoff.

#### 3.1. Mainline Experimental Setup

We first describe our mainline experimental setup. In the next subsection, we vary each design choice to check robustness of the phenomenon. More extensive experimental results are given in the appendix.

For the mainline experiment, we trained a ResNet34 (He et al., 2016) on the CIFAR10 dataset (Krizhevsky et al., 2009). We trained using stochastic gradient descent (SGD) with momentum 0.9. The initial learning rate is 0.1. We applied stage-wise training (decay learning rate by a factor of 10 every 200 epochs), and used weight decay \( 5 \times 10^{-4} \). To change the model complexity of the neural network, we scale the number of filters (i.e., width) of the convolutional layers. More specifically, with width \( w \), the number of filters are \([w, 2w, 4w, 8w] \). We vary \( w \) from 2 to 64 (the width \( w \) of a regular ResNet34 designed for CIFAR10 in He et al. (2016) is 16).

Relative to the standard experimental setup (He et al., 2016), there are two main differences. First, since bias-variance is usually defined for the squared loss (see (1) and (2)), our loss function is the squared error (squared \( \ell_2 \) distance between the softmax probabilities and the one-hot class vector) rather than the log-loss. In the next section we also consider models trained with the log-loss and estimate the bias and variance by using a generalized bias-variance decomposition, as described in §2.2. Second, to measure the variance (and hence bias), we need two models trained on independent subsets of the data as discussed in §2.2. Therefore, the training dataset is split in half and each model is trained on only \( n = 25,000 = 50,000/2 \) data points. We estimate the variance by averaging over \( N = 3 \) such random splits (i.e., we train \( 6 = 3 \times 2 \) copies of each model).

In Figure 2, we can see that the variance as a function of the width is unimodal and the bias is monotonically decreasing. Since the scale of the variance is small relative to the bias, the overall behavior of the risk is monotonically decreasing.

#### 3.2. Varying Architectures, Loss Functions, Datasets

**Architectures.** We observe the same monotonically decreasing bias and unimodal variance phenomenon for ResNext29 (Xie et al., 2017). To scale the “width” of the ResNext29, we first set the number of channels to 1 and increase the cardinality, defined in (Xie et al., 2017), from 2 to 4, and then fix the cardinality at 4 and increase channel...
Figure 2. Mainline experiment on ResNet34, CIFAR10 dataset (25,000 training samples). (Left) Risk, bias, and variance for ResNet34. (Middle) Variance for ResNet34. (Right) Train error and test error for ResNet34.

(a) ResNext29, MSE loss, CIFAR10 (b) ResNet34, CE loss, CIFAR10 (c) DNN, MSE loss, MNIST

Figure 3. Risk, bias, and variance with respect to different network architectures, training loss functions, and datasets. (a). ResNext29 trained by MSE loss on CIFAR10 dataset (25,000 training samples). (b). ResNet34 trained by CE loss (estimated by generalized bias-variance decomposition using Bregman divergence) on CIFAR10 dataset (10,000 training samples). (c). Fully connected network with one hidden layer and ReLU activation trained by MSE loss on MNIST dataset (10,000 training samples).

size from 1 to 32. Results are shown in Figure 3(a), where the width on the x-axis is defined as the cardinality times the filter size.

Loss Function. In addition to the bias-variance decomposition for MSE loss, we also considered a similar decomposition for cross-entropy loss as described in §2.2. We train with cross-entropy loss and use $n = 10,000$ training samples (5 splits), repeating $N = 4$ times with independent random splits. As shown in Figure 3(b), the behavior of the generalized bias and variance for cross entropy is consistent with our earlier observations: the bias is monotonically decreasing and the variance is unimodal. The risk first increases and then decreases, corresponding to the unimodal risk pattern in Figure 1(c).

Datasets. In addition to CIFAR10, we study bias and variance on MNIST (LeCun, 1998) and Fashion-MNIST (Xiao et al., 2017). For these two datasets, we use a fully connected neural network with one hidden layer with ReLU activation function. The “width” of the network is the number of hidden nodes. We use 10,000 training samples ($N = 5$). As seen in Figure 3(c) and 10 (in Appendix B), for both MNIST and Fashion-MNIST, the variance is again unimodal and the bias is monotonically decreasing.

In addition to the above experiments, we also conduct experiments on the CIFAR100 dataset, the VGG network architecture (Simonyan & Zisserman, 2015), various training sample sizes, and different weight decay regularizations and present the results in Appendix B. We observe the same monotonicity decreasing bias and unimodal variance phenomenon in all of these experiments.

3.3. Connection to Double-Descent Risk

When the relative scale of bias and variance changes, the risk displays one of the three patterns, monotonically decreasing, double descent, and unimodal, as presented in Figure 1(a), 1(b) and 1(c). In particular, the recent stream of observations on double descent risk (Belkin et al., 2019a) can be explained by unimodal variance and monotonically decreasing bias. In our experiments, including the experiments in previous sections, we typically observe monotonicity decreasing risk; but with more label noise, the variance will increase and we observe the double descent risk curve.

Label Noise. Similar to the setup in Nakkiiran (2019), for each split, we sample training data from the whole training dataset, and replace the label of each training example with a uniform random class with independent probability $p$. Label noise increases the variance of the model and hence leads to double-descent risk as seen in Figure 4. If the variance is small, the risk does not have the double-descent shape
because the variance peak is not large enough to overwhelm the bias, as observed in Figures 2, 3(a), 3(c) and 10.

### 3.4. Discussion of Possible Sources of Error

In this section, we briefly describe the possible sources of error in our estimator defined in §2.2.

**Mean Squared Error.** As argued in §2.2, the variance estimator is unbiased. To understand the variance of the estimator, we first split the data into two parts. For each part, we compute the bias and variance for varying network width by using our estimator. Averaging across different model width, the relative difference between the two parts is 0.6% for bias and 3% for variance, so our results for MSE are minimally sensitive to finite-sample effects. The complete experiments can be found in the appendix (see Figure 17).

**Cross Entropy Loss.** For cross entropy loss, we are currently unable to obtain an unbiased estimator. We can assess the quality of our estimator using the following scheme. We partition the dataset into five parts \( T_1, \ldots, T_5 \), i.e., set \( N = 5 \) in Algorithm 1. Then, we sequentially plot the estimate of bias and variance using \( k = 1, 2, 3, 4 \) as described in Algorithm 1. We observe that using larger \( k \) gives better estimates. In Figure 18 of Appendix B.9, we observe that as \( k \) increases, the bias curve systematically decreases and the variance curve increases. Therefore our estimator over-estimates the bias and under-estimates the variance, but the overall behaviors of the curves remain consistent.

### 4. What Affects the Bias and Variance?

In this section, through the Bias-Variance decomposition analyzed in §3, we investigate the role of depth for neural networks and the robustness of neural networks on out-of-distribution examples.

#### 4.1. Bias-Variance Tradeoff for Out-of-Distribution (OOD) Example

For many real-world computer vision applications, inputs can be corrupted by random noise, blur, weather, etc. These common occurring corruptions are shown to significantly decrease model performance (Azulay & Weiss, 2019; Hendrycks & Dietterich, 2019). To better understand the “generalization gap” between in-distribution test examples and out-of-distribution test examples, we empirically evaluate the bias and variance on the CIFAR10-C dataset developed by Hendrycks & Dietterich (2019), which is a common corruption benchmark and includes 15 types of corruption.

By applying the models trained in the mainline experiment, we are able to evaluate the bias and variance on CIFAR10-C test dataset according to the definitions in (1) and (2). As we can see from Figure 5(a), both the bias and variance increase relative to the original CIFAR10 test set. Consistent with the phenomenon observed in the mainline experiment, the bias dominates the overall risk. The results indicate that the “generalization gap” mainly comes from increased bias, with relatively less contribution from variance as well.

#### 4.2. Effect of Model Depth on Bias and Variance

In addition to the ResNet34 considered in the mainline experiment, we also evaluate the bias and variance for ResNet18 and ResNet50. Same as the mainline experiment setup, we estimate the bias and variance for ResNet using 25,000 training samples (\( N = 2 \)) and three independent random splits (\( k = 3 \)). The standard building block of ResNet50 architecture in He et al. (2016) is bottleneck block, which is different from the basic block used in ResNet18 and ResNet34. To ensure that depth is the only changing variable across three architectures, we apply the basic block for ResNet50. Same training epochs and learning rate decays...
From Figure 5(b) and 5(c), we observe that the bias decreases as the depth increases, while the variance increases as the depth increases. For each model, the bias is monotonically decreasing and the variance is unimodal. The differences in variance are small (around 0.01) compared with the changes in bias. Overall, the risk typically decreases as the depth increases. Our experimental results suggest that the improved generalization for deeper models, with the same network architecture, are mainly attributed to lower bias.

For completeness, we also include the bias and variance versus depth when basic blocks in ResNet are replaced by bottleneck blocks (see Figure 20 in the appendix). We observe similar qualitative trend of bias and variance.

Note that at high width, the bias of ResNet50 is slightly higher than the bias of ResNet18 and ResNet34. We attribute this inconsistency to difficulties when training ResNet50 without bottleneck blocks at high width. Lastly, we also include the bias and variance versus depth for out-of-distribution test samples, in which case we also observed decreased bias and increased variance as depth increases, as shown in Figure 19 of Appendix B.10.

5. Theoretical Insights from a Two-layer Linear Model

While the preceding experiments show that the bias and variance robustly exhibit monotonic-unimodal behavior in the random-design setting, existing theoretical analyses hold instead for the fixed-design setting, where the behavior of the bias and variance are more complex, with both the bias and variance exhibiting a peak and the risk exhibiting double descent pattern (Mei & Montanari (2019, Figure 6)). In general, while the risk should be the same (in expectation) for the random and fixed design setting, the fixed-design setting has lower bias and higher variance.

Motivated by the more natural behavior in the random-design setting, we work to extend the existing fixed-design theory to the random-design case. Our starting point is Mei & Montanari (2019), who consider two-layer non-linear networks with random hidden layer weights. However, the randomness in the design complicates the analysis, so we make two points of departure to help simplify: first, we consider two-layer linear rather than non-linear networks, and second, we consider a different scaling limit ($n/d \rightarrow \infty$ rather than $n/d$ going to some constant). In this setting, we rigorously show that the variance is indeed unimodal and the bias is monotonically decreasing (Figure 6). Our precise assumptions are given below.

5.1. Model Assumptions

We consider the task of learning a function $y = f(x)$ that maps each input vector $x \in \mathbb{R}^d$ to an output (label) value $y \in \mathbb{R}$. The input-output pair $(x, y)$ is assumed to be drawn from a distribution where $x \sim \mathcal{N}(0, I_{d/d})$ and

$$y = f_0(x) := x^\top \theta,$$

where $\theta \in \mathbb{R}^d$ is a weight vector. Given a training set $T := \{(x_i, y_i)\}_{i=1}^n$ with training samples drawn independently from the data distribution, we learn a two-layer linear neural network parametrized by $W \in \mathbb{R}^{p \times d}$ and $\beta \in \mathbb{R}^p$ as

$$f(x) = (W x)^\top \beta,$$

where $p$ is the number of hidden units in the network. In above, we take $W$ as a parameter independent of the training data $T$ whose entries are drawn from i.i.d. Gaussian distribution $\mathcal{N}(0, 1/d)$. Given $W$, the parameter $\beta$ is estimated by solving the following ridge regression\footnote{ regularization on weight parameters is arguably the most widely used technique in training neural network, known for improving generalization (Krogh & Hertz, 1992). Other regularization such as $\ell_1$ can also be used and leads to qualitatively similar behaviors.} problem

$$\beta_\lambda(T, W) = \arg\min_{\beta \in \mathbb{R}^p} \| (W X) \beta - y \|_2^2 + \lambda \| \beta \|_2^2,$$
We may now calculate the bias and variance of the model $f$. The asymptotic expressions for bias and variance are parametrized by the dimension of the input $d$. We derive the precise expression for the bias and variance of the model. From our expression, we obtain

\[
\beta_\lambda(T, W) = (W X X^T + \lambda I)^{-1} W X y,
\]

for different triples. We assume that there exists a $\gamma > 0$ such that

\[
\lim_{d \to \infty} \frac{p(d)}{d} = \gamma, \quad \text{and} \quad \lim_{d \to \infty} \frac{n(d)}{d} = \infty.
\]

For simplicity, we will write $n := n(d)$ and $p := p(d)$.

With the assumption above, we have the expression of the expected bias and variance as a function of $\gamma$ and $\lambda$.

**Theorem 1.** Given $\{(d, n(d), p(d))\}_{d=1}^\infty$ that satisfies Assumption 1, let $\lambda = \frac{p}{d} \lambda_0$ for some fixed $\lambda_0 > 0$. The asymptotic expression of expected bias and variance are given by

\[
\lim_{d \to \infty} \text{Bias}^2_\lambda = \frac{1}{4} \Phi_3(\lambda_0, \gamma),
\]

\[
\lim_{d \to \infty} \text{Variance}_\lambda = \begin{cases} 
\Phi_1(\lambda_0, \frac{1}{2}) & \frac{1}{2} \Phi_3(\lambda_0, \frac{1}{2}) - \frac{(1-\gamma)(1-2\gamma)}{2\gamma} - \frac{1}{4} \Phi_3(\lambda_0, \gamma)^2, \quad \gamma \leq 1, \\
\Phi_1(\lambda_0, \gamma) - \frac{\gamma-1}{2} - \frac{1}{4} \Phi_3(\lambda_0, \gamma)^2, & \gamma > 1,
\end{cases}
\]

where

\[
\Phi_1(\lambda_0, \gamma) = \lambda_0(\gamma + 1) + (\gamma - 1)^2,
\]

\[
\Phi_2(\lambda_0, \gamma) = \sqrt{(\lambda_0 + 1)^2 + 2(\lambda_0 - 1)\gamma + \gamma^2},
\]

\[
\Phi_3(\lambda_0, \gamma) = \Phi_2(\lambda_0, \gamma) - \lambda_0 - \gamma + 1.
\]

The proof is given in Appendix C.

The risk can be obtained through $\text{Bias}^2_\lambda + \text{Variance}_\lambda$. The expression in Theorem 1 is plotted as the red curves in Figure 6. In addition to the case when $n/d \to \infty$, we also plot the shape of bias, variance and risk when $n/d \to \{0.15, 0.25, 0.35, \ldots, 1.00, 1.50\}$. We find that the risk of the model grows from unimodal to monotonically decreasing as the number of samples increased (see Figure...
Moreover, the bias of the model is monotonically decreasing (see Figure 6(b)) and the variance is unimodal (see Figure 6(c)).

**Corollary 1 (Monotonicity of Bias).** The derivative of the limiting expected bias in (9) can be calculated as

\[
\frac{d}{d\lambda} \operatorname{Bias}_\lambda = \frac{\sqrt{2(\gamma + 1)\lambda_0 + (\gamma - 1)^2 + \lambda_0^2} - \gamma - \lambda_0 + 1}{2\sqrt{\gamma^2 + 2\gamma(\lambda_0 - 1) + (\lambda_0 + 1)^2}}.
\]

When \(\lambda_0 \geq 0\), the expression in (10) is strictly non-positive, therefore the limiting expected bias is monotonically non-increasing as a function of \(\gamma\), as classical theories predicts.

To gain further insight into the above formulas, we also consider the case when the ridge regularization amount \(\lambda_0\) is small. In particular, we consider the first order effect of \(\lambda_0\) on the bias and variance term, and compute the value of \(\gamma\) where the variance attains the peak.

**Corollary 2 (Unimodality of Variance – small \(\lambda_0\) limit).** Under the assumptions of Theorem 1, the first order effect of \(\lambda_0\) on variance is given by

\[
\lim_{d \to \infty} \mathbb{E} \operatorname{Variance}_\lambda = \begin{cases} 
O(\lambda_0^2), & \gamma > 1, \\
-(\gamma - 1)\gamma - 2\lambda_0 + O(\lambda_0^2), & \text{o.w.}
\end{cases}
\]

and the risk is given by

\[
\lim_{d \to \infty} \mathbb{E} \operatorname{Risk}_\lambda = \begin{cases} 
1 - \gamma + O(\lambda_0^2), & \gamma \leq 1, \\
O(\lambda_0^2), & \gamma > 1.
\end{cases}
\]

Moreover, up to first order, the peak in the variance is

\[
\text{Peak} = \frac{1}{2} - \lambda_0 + O(\lambda_0^2).
\]

Theorem 2 suggests that when \(\lambda_0\) is sufficiently small, the variance of the model is maximized when \(p = d/2\), and the effect of \(\lambda_0\) is to shift the peak slightly to \(d/2 - \lambda_0 d\).

From a technical perspective, to compute the variance in the random-design setting, we need to compute the element-wise expectation of certain random matrix. For this purpose, we apply the combinatorics of counting non-cross partitions to characterize the asymptotic expectation of products of Wishart matrices.

### 6. Conclusion and Discussion

In this paper we re-examine the classical theory of bias and variance trade-off as the width of a neural network increases. Through extensive experimentation, our main finding is that, while the bias is monotonically decreasing as classical theory would predict, the variance is unimodal. This combination leads to three typical risk curve patterns, all observed in practice. Theoretical analysis of a two-layer linear network corroborates these experimental observations.

The seemingly varied and baffling behaviors of modern neural networks are thus in fact consistent, and explainable through classical bias-variance analysis. The main unexplained mystery is the unimodality of the variance. We conjecture that as the model complexity approaches and then goes beyond the data dimension, it is regularization in model estimation (the ridge penalty in our theoretical example) that helps bring down the variance. Under this account, the decrease in variance for large dimension comes from better conditioning of the empirical covariance, making it better-aligned with the regularizer.

In the future, it would be interesting to see if phenomena characterized by the simple two-layer model can be rigorously generalized to deeper networks with nonlinear activation, probably revealing other interplays between model complexity and regularization (explicit or implicit). Such a study could also help explain another phenomenon we (and others) have observed: bias decreases with more layers as variance increases. We believe that the (classic) bias-variance analysis remains a powerful and insightful framework for understanding the behaviors of deep networks; properly used, it can guide practitioners to design more generalizable and robust networks in the future.

**Acknowledgements.** We would like to thank Emmanuel Candés for first bringing the double-descent phenomenon to our attention, Song Mei for helpful discussion regarding random v.s. fixed design regression, and Nikhil Srivastava for pointing out to relevant references in random matrix theory. We would also like to thank Preetum Nakkiran, Mihaela Curmei, and Chloe Hsu for valuable feedback during preparation of this manuscript. The authors acknowledge support from Tsinghua-Berkeley Shenzhen Institute Research Fund and BAIR.

### References


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,


