DEEP DOUBLE DESCENT: WHERE BIGGER MODELS AND MORE DATA HURT

Preetum Nakkiran[*] Harvard University
Gal Kaplun[†] Harvard University
Yamini Bansal[†] Harvard University
Tristan Yang Harvard University
Boaz Barak Harvard University
Ilya Sutskever OpenAI

ABSTRACT

We show that a variety of modern deep learning tasks exhibit a "double-descent" phenomenon where, as we increase model size, performance first gets *worse* and then gets better. Moreover, we show that double descent occurs not just as a function of model size, but also as a function of the number of training epochs. We unify the above phenomena by defining a new complexity measure we call the *effective model complexity* and conjecture a generalized double descent with respect to this measure. Furthermore, our notion of model complexity allows us to identify certain regimes where increasing (even quadrupling) the number of train samples actually *hurts* test performance.

1 INTRODUCTION

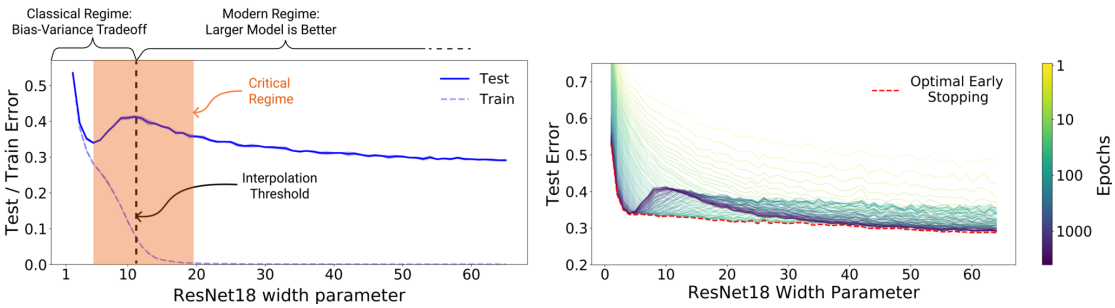


Figure 1: **Left:** Train and test error as a function of model size, for ResNet18s of varying width on CIFAR-10 with 15% label noise. **Right:** Test error, shown for varying train epochs. All models trained using Adam for 4K epochs. The largest model (width 64) corresponds to standard ResNet18.

The *bias-variance trade-off* is a fundamental concept in classical statistical learning theory (e.g., Hastie et al. (2005)). The idea is that models of higher complexity have lower bias but higher variance. According to this theory, once model complexity passes a certain threshold, models "overfit" with the variance term dominating the test error, and hence from this point onward, increasing model complexity will only *decrease* performance (i.e., increase test error). Hence conventional wisdom in classical statistics is that, once we pass a certain threshold, *"larger models are worse."*

However, modern neural networks exhibit no such phenomenon. Such networks have millions of parameters, more than enough to fit even random labels (Zhang et al. (2016)), and yet they perform much better on many tasks than smaller models. Indeed, conventional wisdom among practitioners is that *"larger models are better"* (Krizhevsky et al. (2012), Huang et al. (2018), Szegedy et al.

[*]Work performed in part while Preetum Nakkiran was interning at OpenAI, with Ilya Sutskever. We especially thank Mikhail Belkin and Christopher Olah for helpful discussions throughout this work. Correspondence Email: preetum@cs.harvard.edu

[†]Equal contribution

深度双重下降：更大的模型和更多的数据有害

普里图姆·纳基兰* 哈佛大学
加尔·卡普伦† 哈佛大学
Yamini Bansal† 哈佛大学
杨特里斯坦 哈佛大学
巴拉克·博阿兹 哈佛大学
Ilya Sutskever OpenAI

摘要

我们表明，各种现代深度学习任务表现出“双重下降”现象，即随着模型规模的增加，性能首先变得*worse*，然后变得更好。此外，我们表明双重下降不仅是模型规模的函数，还与训练周期数有关。我们通过定义一个新的复杂性度量来统一上述现象，我们称之为*effective model complexity*，并推测相对于该度量的广义双重下降。此外，我们的模型复杂性概念使我们能够识别某些情况下，增加（甚至四倍增加）训练样本数量实际上*hurts*测试性能。

1 介绍

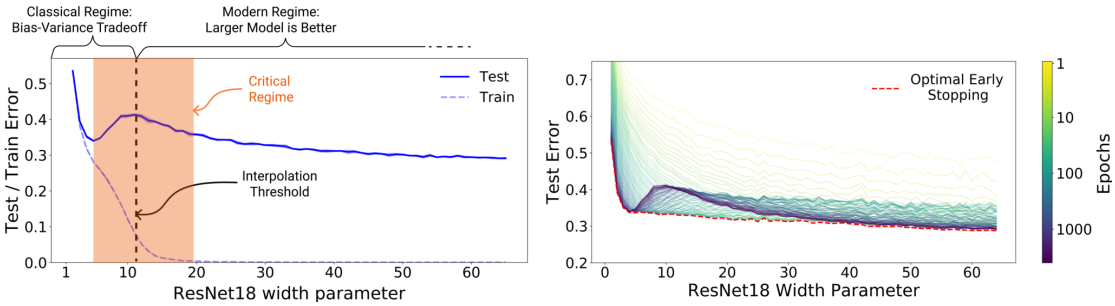


图 1：左图：在 CIFAR-10 上具有 15% 标签噪声的不同宽度的 ResNet18 的模型大小与训练和测试误差的关系。右图：测试误差，显示了不同训练周期的情况。所有模型均使用 Adam 训练 4K 轮。最大模型（宽度 64）对应于标准 ResNet18。

bias-variance trade-off 是经典统计学习理论中的一个基本概念（例如，Hastie 等人 (2005)）。其思想是，复杂度更高的模型具有更低的偏差但更高的方差。根据该理论，一旦模型复杂度超过某个阈值，模型就会“过拟合”，方差项主导测试误差，因此从这一点开始，增加模型复杂度只会 *decrease* 性能（即增加测试误差）。因此，经典统计学中的传统观点是，一旦我们超过某个阈值，*“larger models are worse.”*

然而，现代神经网络并没有表现出这种现象。这些网络拥有数百万个参数，足以拟合甚至是随机标签（Zhang et al. (2016)），但在许多任务上它们的表现远远优于较小的模型。实际上，实践者中的传统观点是*“larger models are better”*（Krizhevsky et al. (2012)，Huang et al. (2018)，Szegedy et al.）。

*Work performed in part while Preetum Nakkiran was interning at OpenAI, with Ilya Sutskever. We especially thank Mikhail Belkin and Christopher Olah for helpful discussions throughout this work. Correspondence Email: preetum@cs.harvard.edu

†Equal contribution

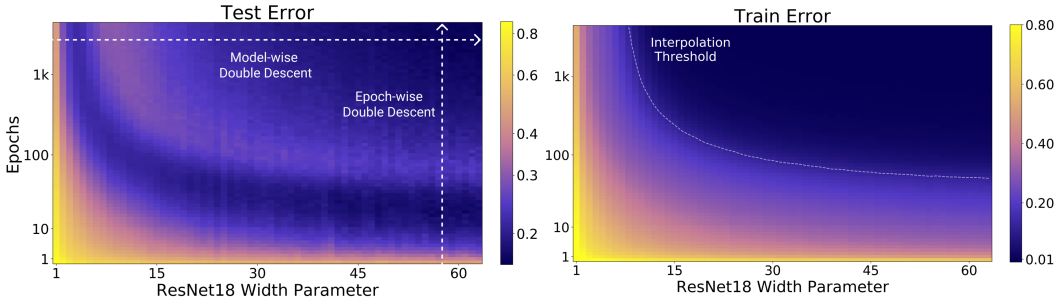


Figure 2: **Left:** Test error as a function of model size and train epochs. The horizontal line corresponds to model-wise double descent–varying model size while training for as long as possible. The vertical line corresponds to epoch-wise double descent, with test error undergoing double-descent as train time increases. **Right** Train error of the corresponding models. All models are Resnet18s trained on CIFAR-10 with 15% label noise, data-augmentation, and Adam for up to 4K epochs.

(2015), Radford et al. (2019)). The effect of training time on test performance is also up for debate. In some settings, "early stopping" improves test performance, while in other settings training neural networks to zero training error only improves performance. Finally, if there is one thing both classical statisticians and deep learning practitioners agree on is *"more data is always better"*.

In this paper, we present empirical evidence that both reconcile and challenge some of the above "conventional wisdoms." We show that many deep learning settings have two different regimes. In the *under-parameterized* regime, where the model complexity is small compared to the number of samples, the test error as a function of model complexity follows the U-like behavior predicted by the classical bias/variance tradeoff. However, once model complexity is sufficiently large to *interpolate* i.e., achieve (close to) zero training error, then increasing complexity only *decreases* test error, following the modern intuition of "bigger models are better". Similar behavior was previously observed in Opper (1995; 2001), Advani & Saxe (2017), Spigler et al. (2018), and Geiger et al. (2019b). This phenomenon was first postulated in generality by Belkin et al. (2018) who named it "double descent", and demonstrated it for decision trees, random features, and 2-layer neural networks with ℓ_2 loss, on a variety of learning tasks including MNIST and CIFAR-10.

Main contributions. We show that double descent is a robust phenomenon that occurs in a variety of tasks, architectures, and optimization methods (see Figure 1 and Section 5; our experiments are summarized in Table A). Moreover, we propose a much more general notion of "double descent" that goes beyond varying the number of parameters. We define the *effective model complexity (EMC)* of a training procedure as the maximum number of samples on which it can achieve close to zero training error. The EMC depends not just on the data distribution and the architecture of the classifier but also on the training procedure—and in particular increasing training time will increase the EMC.

We hypothesize that for many natural models and learning algorithms, double descent occurs as a function of the EMC. Indeed we observe "epoch-wise double descent" when we keep the model fixed and increase the training time, with performance following a classical U-like curve in the underfitting stage (when the EMC is smaller than the number of samples) and then improving with training time once the EMC is sufficiently larger than the number of samples (see Figure 2). As a corollary, early stopping only helps in the relatively narrow parameter regime of critically parameterized models.

Sample non-monotonicity. Finally, our results shed light on test performance as a function of the number of train samples. Since the test error peaks around the point where EMC matches the number of samples (the transition from the under- to over-parameterization), increasing the number of samples has the effect of shifting this peak to the right. While in most settings increasing the number of samples decreases error, this shifting effect can sometimes result in a setting where *more data is worse!* For example, Figure 3 demonstrates cases in which increasing the number of samples by a factor of 4.5 results in worse test performance.

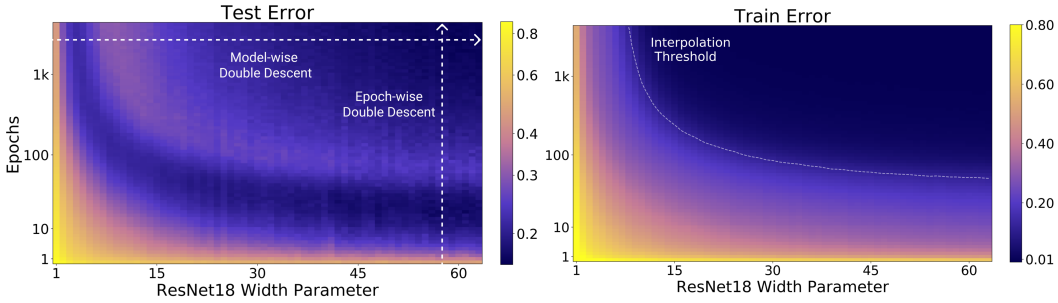


图2：左图：测试误差作为模型大小和训练周期的函数。水平线对应于模型层面的双重下降——在尽可能长的训练时间内改变模型大小。垂直线对应于周期层面的双重下降，随着训练时间的增加，测试误差经历双重下降。右图：对应模型的训练误差。所有模型都是在CIFAR-10上训练的Resnet18，标签噪声为15%，使用数据增强和Adam优化器，训练时间最长为4000个周期。

(2015)，Radford 等人 (2019))。训练时间对测试性能的影响也存在争议。在某些情况下，“提前停止”可以提高测试性能，而在其他情况下，将神经网络训练到零训练误差只会提高性能。最后，如果有一件事是经典统计学家和深度学习从业者都同意的，那就是*“more data is always better”*。

在本文中，我们提供了实证证据，既调和又挑战了一些上述“传统智慧”。我们展示了许多深度学习设置中存在两种不同的模式。在*under-parameterized*模式中，当模型复杂度相对于样本数量较小时，测试误差作为模型复杂度的函数遵循经典偏差/方差权衡预测的U型行为。然而，一旦模型复杂度足够大达到*interpolate*，即实现（接近）零训练误差时，增加复杂度只会*decreases*测试误差，遵循现代“更大模型更好”的直觉。类似的行为在Opper (1995; 2001)、Advani & Saxe (2017)、Spigler et al. (2018)和Geiger et al. (2019b)中也曾观察到。这一现象最早由Belkin et al. (2018)在一般性中提出，他们将其命名为“双重下降”，并在包括MNIST和CIFAR-10在内的各种学习任务中，针对决策树、随机特征和具有ℓ_2损失的两层神经网络进行了演示。

主要贡献。我们表明，双重下降是一种稳健的现象，发生在各种任务、架构和优化方法中（见图1和第5节；我们的实验总结在表A中）。此外，我们提出了一种更为普遍的“双重下降”概念，这超越了参数数量的变化。我们将训练过程的*effective model complexity (EMC)*定义为其能够在接近零训练误差的情况下处理的最大样本数量。EMC不仅取决于数据分布和分类器的架构，还取决于训练过程——特别是增加训练时间将增加EMC。

我们假设，对于许多自然模型和学习算法，双重下降作为EMC的函数发生。确实，当我们保持模型不变并增加训练时间时，我们观察到“逐周期双重下降”，在欠拟合阶段（当EMC小于样本数量时），性能遵循经典的U型曲线，然后在EMC足够大于样本数量时，随着训练时间的增加而改善（见图2）。作为推论，早停仅在临界参数化模型的相对狭窄的参数范围内有帮助。

样本非单调性。最后，我们的结果揭示了测试性能与训练样本数量之间的关系。由于测试误差在EMC与样本数量匹配的点附近达到峰值（从欠参数化到过参数化的过渡），增加样本数量的效果是将这一峰值向右移动。虽然在大多数情况下增加样本数量会减少误差，但这种移动效应有时会导致一种情况，其中*more data is worse!* 例如，图3展示了将样本数量增加4.5倍导致测试性能变差的情况。

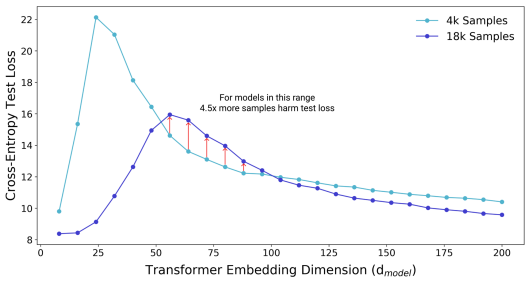


Figure 3: Test loss (per-token perplexity) as a function of Transformer model size (embedding dimension d_{model}) on language translation (IWSLT'14 German-to-English). The curve for 18k samples is generally lower than the one for 4k samples, but also shifted to the right, since fitting 18k samples requires a larger model. Thus, for some models, the performance for 18k samples is *worse* than for 4k samples.

2 Our results

To state our hypothesis more precisely, we define the notion of *effective model complexity*. We define a *training procedure* $\mathcal{T}$ to be any procedure that takes as input a set $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ of labeled training samples and outputs a classifier $\mathcal{T}(S)$ mapping data to labels. We define the *effective model complexity* of $\mathcal{T}$ (w.r.t. distribution $\mathcal{D}$) to be the maximum number of samples n on which $\mathcal{T}$ achieves on average ≈ 0 *training error*.

Definition 1 (Effective Model Complexity) *The* Effective Model Complexity *(EMC) of a training procedure $\mathcal{T}$, with respect to distribution $\mathcal{D}$ and parameter $\epsilon > 0$, is defined as:*

$$\mathrm{EMC}_{\mathcal{D},\epsilon}(\mathcal{T}) := \max\{n \mid \mathbb{E}_{S\sim\mathcal{D}^n}[\mathrm{Error}_S(\mathcal{T}(S))] \leq \epsilon\}$$

where $\mathrm{Error}_S(M)$ is the mean error of model M on train samples S.

Our main hypothesis can be informally stated as follows:

Hypothesis 1 (Generalized Double Descent hypothesis, informal) *For any natural data distribution $\mathcal{D}$, neural-network-based training procedure $\mathcal{T}$, and small $\epsilon > 0$, if we consider the task of predicting labels based on n samples from $\mathcal{D}$ then:*

Under-paremeterized regime. *If $\mathrm{EMC}_{\mathcal{D},\epsilon}(\mathcal{T})$ is sufficiently smaller than n, any perturbation of $\mathcal{T}$ that increases its effective complexity will decrease the test error.*

Over-parameterized regime. *If $\mathrm{EMC}_{\mathcal{D},\epsilon}(\mathcal{T})$ is sufficiently larger than n, any perturbation of $\mathcal{T}$ that increases its effective complexity will decrease the test error.*

Critically parameterized regime. *If $\mathrm{EMC}_{\mathcal{D},\epsilon}(\mathcal{T}) \approx n$, then a perturbation of $\mathcal{T}$ that increases its effective complexity might decrease* **or increase** *the test error.*

Hypothesis 1 is informal in several ways. We do not have a principled way to choose the parameter ϵ (and currently heuristically use $\epsilon = 0.1$). We also are yet to have a formal specification for "sufficiently smaller" and "sufficiently larger". Our experiments suggest that there is a *critical interval* around the *interpolation threshold* when $\mathrm{EMC}_{\mathcal{D},\epsilon}(\mathcal{T}) = n$: below and above this interval increasing complexity helps performance, while within this interval it may hurt performance. The width of the critical interval depends on both the distribution and the training procedure in ways we do not yet completely understand.

We believe Hypothesis 1 sheds light on the interaction between optimization algorithms, model size, and test performance and helps reconcile some of the competing intuitions about them. The main result of this paper is an experimental validation of Hypothesis 1 under a variety of settings, where we considered several natural choices of datasets, architectures, and optimization algorithms, and we changed the "interpolation threshold" by varying the number of model parameters, the length of training, the amount of label noise in the distribution, and the number of train samples.

Model-wise Double Descent. In Section 5, we study the test error of models of increasing size, for a fixed large number of optimization steps. We show that "model-wise double-descent" occurs for various modern datasets (CIFAR-10, CIFAR-100, IWSLT'14 de-en, with varying amounts of label noise), model architectures (CNNs, ResNets, Transformers), optimizers (SGD, Adam), number

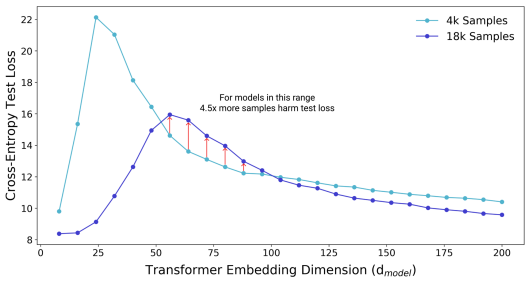


图3：测试损失（每个标记的困惑度）作为Transformer模型大小（嵌入维度d_{model}）在语言翻译（IWSLT‘14德语到英语）中的函数。18k样本的曲线通常低于4k样本的曲线，但也向右移动，因为拟合18k样本需要更大的模型。因此，对于某些模型，18k样本的性能比4k样本的性能是*worse*。

2 我们的结果

为了更精确地陈述我们的假设，我们定义*effective model complexity*的概念。我们将*training procedure* $\mathcal{T}$定义为任何以一组标记的训练样本$S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$为输入并输出一个将数据映射到标签的分类器$\mathcal{T}(S)$的过程。我们定义相对于分布$\mathcal{D}$的$\mathcal{T}$ (的*effective model complexity*为n的最大样本数，在这些样本上$\mathcal{T}$平均达到≈ 0 *training error*。

定义 1（有效模型复杂度）*The* 有效模型复杂度 *(EMC) of a training procedure* $\mathcal{T}$*, with respect to distribution* $\mathcal{D}$ *and parameter* $\epsilon > 0$*, is defined as:*

$$\mathrm{EMC}_{\mathcal{D},\epsilon}(\mathcal{T}) := \max\{n \mid \mathbb{E}_{S\sim\mathcal{D}^n}[\mathrm{Error}_S(\mathcal{T}(S))] \leq \epsilon\}$$

where 错误$_S(M)$ *is the mean error of model* M *on train samples* S*.*

我们的主要假设可以非正式地表述如下：

假设 1（广义双重下降假设，非正式）*For any natural data distribution* $\mathcal{D}$*, neural-network-based training procedure* $\mathcal{T}$*, and small* $\epsilon > 0$*, if we consider the task of predicting labels based on* n *samples from* $\mathcal{D}$ *then:*

欠参数化状态。*If* $\mathrm{EMC}_{\mathcal{D},\epsilon}(\mathcal{T})$ *is sufficiently smaller than* n*, any perturbation of* $\mathcal{T}$ *that increases its effective complexity will decrease the test error.*

过参数化状态。*If* $\mathrm{EMC}_{\mathcal{D},\epsilon}(\mathcal{T})$ *is sufficiently larger than* n*, any perturbation of* $\mathcal{T}$ *that increases its effective complexity will decrease the test error.*

临界参数化状态。*If* $\mathrm{EMC}_{\mathcal{D},\epsilon}(\mathcal{T}) \approx n$*, then a perturbation of* $\mathcal{T}$ *that increases its effective complexity might decrease* 或增加 *the test error.*

假设 1 在几个方面是不正式的。我们没有一个有原则的方法来选择参数 ϵ (，目前凭经验使用 $\epsilon = 0.1$)。我们也尚未对“足够小”和“足够大”进行正式规范。我们的实验表明，当 $\mathrm{EMC}_{\mathcal{D},\epsilon}(\mathcal{T}) = n$ 时，*critical interval* 在 *interpolation threshold* 附近：在这个区间的下方和上方，增加复杂性有助于性能，而在这个区间内可能会损害性能。关键区间的宽度取决于分布和训练过程的方式，而这些方式我们尚未完全理解。

我们认为假设1揭示了优化算法、模型大小和测试性能之间的相互作用，并有助于调和关于它们的一些相互竞争的直觉。本文的主要结果是在各种设置下对假设1的实验验证，其中我们考虑了几种自然选择的数据集、架构和优化算法，并通过改变模型参数的数量、训练的长度、分布中的标签噪声量以及训练样本的数量来改变“插值阈值”。

模型层面的双重下降。在第5节中，我们研究了在固定的大量优化步骤下，随着模型规模增加的测试误差。我们展示了“模型层面的双重下降”在各种现代数据集（CIFAR-10、CIFAR-100、IWSLT‘14 de-en，具有不同程度的标签噪声）、模型架构（CNNs、ResNets、Transformers）、优化器（SGD、Adam）、数量上发生。

of train samples, and training procedures (data-augmentation, and regularization). Moreover, the peak in test error systematically occurs at the interpolation threshold. In particular, we demonstrate realistic settings in which *bigger models are worse*.

Epoch-wise Double Descent. In Section 6, we study the test error of a fixed, large architecture over the course of training. We demonstrate, in similar settings as above, a corresponding peak in test performance when models are trained just long enough to reach ≈ 0 train error. The test error of a large model first decreases (at the beginning of training), then increases (around the critical regime), then decreases once more (at the end of training)—that is, *training longer can correct overfitting.*

Sample-wise Non-monotonicity. In Section 7, we study the test error of a fixed model and training procedure, for varying number of train samples. Consistent with our generalized double-descent hypothesis, we observe distinct test behavior in the "critical regime", when the number of samples is near the maximum that the model can fit. This often manifests as a long plateau region, in which taking significantly more data might not help when training to completion (as is the case for CNNs on CIFAR-10). Moreover, we show settings (Transformers on IWSLT'14 en-de), where this manifests as a peak—and for a fixed architecture and training procedure, *more data actually hurts.*

Remarks on Label Noise. We observe all forms of double descent most strongly in settings with label noise in the train set (as is often the case when collecting train data in the real-world). However, we also show several realistic settings with a test-error peak even without label noise: ResNets (Figure 4a) and CNNs (Figure 20) on CIFAR-100; Transformers on IWSLT'14 (Figure 8). Moreover, all our experiments demonstrate distinctly different test behavior in the critical regime— often manifesting as a "plateau" in the test error in the noiseless case which develops into a peak with added label noise. See Section 8 for further discussion.

3 Related work

Model-wise double descent was first proposed as a general phenomenon by Belkin et al. (2018). Similar behavior had been observed in Opper (1995; 2001), Advani & Saxe (2017), Spigler et al. (2018), and Geiger et al. (2019b). Subsequently, there has been a large body of work studying the double descent phenomenon. A growing list of papers that theoretically analyze it in the tractable setting of linear least squares regression includes Belkin et al. (2019); Hastie et al. (2019); Bartlett et al. (2019); Muthukumar et al. (2019); Bibas et al. (2019); Mitra (2019); Mei & Montanari (2019). Moreover, Geiger et al. (2019a) provide preliminary results for model-wise double descent in convolutional networks trained on CIFAR-10. Our work differs from the above papers in two crucial aspects: First, we extend the idea of double-descent beyond the number of parameters to incorporate the training procedure under a unified notion of "Effective Model Complexity", leading to novel insights like epoch-wise double descent and sample non-monotonicity. The notion that increasing train time corresponds to increasing complexity was also presented in Nakkiran et al. (2019). Second, we provide an extensive and rigorous demonstration of double-descent for modern practices spanning a variety of architectures, datasets optimization procedures. An extended discussion of the related work is provided in Appendix C.

4 Experimental Setup

We briefly describe the experimental setup here; full details are in Appendix B [1]. We consider three families of architectures: ResNets, standard CNNs, and Transformers. **ResNets:** We parameterize a family of ResNet18s (He et al. (2016)) by scaling the width (number of filters) of convolutional layers. Specifically, we use layer widths $[k, 2k, 4k, 8k]$ for varying k. The standard ResNet18 corresponds to $k = 64$. **Standard CNNs:** We consider a simple family of 5-layer CNNs, with 4 convolutional layers of widths $[k, 2k, 4k, 8k]$ for varying k, and a fully-connected layer. For context, the CNN with width $k = 64$, can reach over 90% test accuracy on CIFAR-10 with data-augmentation. **Transformers:** We consider the 6 layer encoder-decoder from Vaswani et al. (2017), as implemented by Ott et al. (2019). We scale the size of the network by modifying the embedding dimension d_{model}, and setting the width of the fully-connected layers proportionally ($d_{\text{ff}} = 4 \cdot d_{\text{model}}$).

[1] The raw data from our experiments are available at: `https://gitlab.com/harvard-machine-learning/double-descent/tree/master`

的训练样本和训练过程（数据增强和正则化）。此外，测试误差的峰值系统地出现在插值阈值处。特别是，我们展示了在其中 *bigger models are worse* 的现实设置。

历元式双重下降。在第6节中，我们研究了在训练过程中固定的大型架构的测试误差。我们展示了在类似的设置中，当模型训练时间恰好足够达到≈ 0训练误差时，测试性能出现相应的峰值。大型模型的测试误差首先减少（在训练开始时），然后增加（在关键阶段附近），最后再次减少（在训练结束时）——即*training longer can correct overfitting.*。

样本非单调性。在第7节中，我们研究了固定模型和训练过程的测试误差，针对不同数量的训练样本。与我们广义的双重下降假设一致，我们观察到在“临界状态”下的不同测试行为，当样本数量接近模型可以拟合的最大值时。这通常表现为一个长平台区域，在该区域中，即使显著增加数据量，在训练完成时可能也无济于事（如在CIFAR-10上的CNN情况）。此外，我们展示了一些设置（IWSLT‘14 en-de上的Transformers），在这些设置中，这种现象表现为一个峰值——对于固定的架构和训练过程，*more data actually hurts.*。

关于标签噪声的备注。我们观察到，在训练集存在标签噪声的情况下，各种形式的双重下降现象最为明显（这在现实世界中收集训练数据时常常发生）。然而，我们也展示了几个即使没有标签噪声也会出现测试误差峰值的现实情况：在CIFAR-100上的ResNets（图4a）和CNNs（图20）；在IWSLT‘14上的Transformers（图8）。此外，我们的所有实验都展示了在关键状态下截然不同的测试行为——在无噪声情况下通常表现为测试误差的“平台”，而在加入标签噪声后发展为峰值。有关进一步讨论，请参见第8节。

3 相关工作

模型双重下降现象最早由 Belkin 等人（2018）提出。类似的行为在 Opper（1995; 2001）、Advani & Saxe（2017）、Spigler 等人（2018）和 Geiger 等人（2019b）中也有观察到。随后，大量研究工作开始研究双重下降现象。在线性最小二乘回归的可处理环境中，理论分析这一现象的论文不断增加，包括 Belkin 等人（2019）；Hastie 等人（2019）；Bartlett 等人（2019）；Muthukumar 等人（2019）；Bibas 等人（2019）；Mitra（2019）；Mei & Montanari（2019）。此外，Geiger 等人（2019a）为在 CIFAR-10 上训练的卷积网络中的模型双重下降提供了初步结果。我们的工作在两个关键方面与上述论文不同：首先，我们将双重下降的概念从参数数量扩展到在统一的“有效模型复杂性”概念下结合训练过程，带来了新的见解，如逐轮双重下降和样本非单调性。Nakkiran 等人（2019）也提出了增加训练时间对应于增加复杂性的观点。其次，我们为现代实践提供了广泛而严格的双重下降演示，涵盖了各种架构、数据集和优化过程。相关工作的扩展讨论见附录 C。

4 实验设置

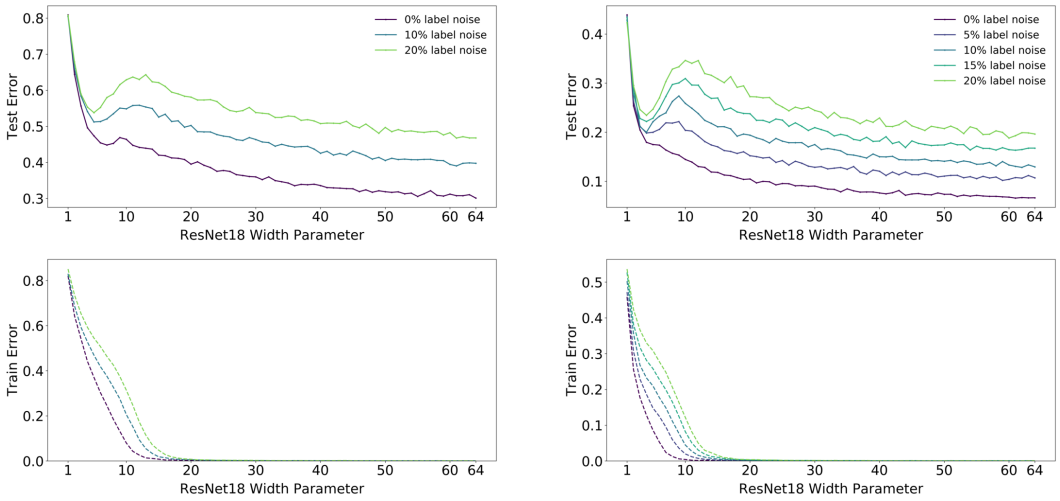
我们在此简要描述实验设置；完整细节见附录B [1]。我们考虑三类架构：ResNets、标准CNN和Transformers。ResNets：我们通过缩放卷积层的宽度（滤波器数量）来参数化一组ResNet 18（He等人（2016））。具体来说，我们使用层宽[k, $2k$, $4k$, $8k$]来变化k。标准ResNet18对应于$k = 64$。标准CNN：我们考虑一个简单的5层CNN家族，具有4个卷积层，宽度为[k, $2k$, $4k$, $8k$]，用于变化k，以及一个全连接层。作为参考，宽度为$k = 64$的CNN在使用数据增强的情况下，可以在CIFAR-10上达到超过90%的测试准确率。Transformers：我们考虑Vaswani等人（2017）提出的6层编码器-解码器，由Ott等人（2019）实现。我们通过修改嵌入维度d_{model}来缩放网络的大小，并成比例地设置全连接层的宽度（$d_{\text{ff}} = 4 \cdot d_{\text{model}}$）。

[1] The raw data from our experiments are available at: `https://gitlab.com/harvard-machine-learning/double-descent/tree/master`

For ResNets and CNNs, we train with cross-entropy loss, and the following optimizers: (1) Adam with learning-rate 0.0001 for 4K epochs; (2) SGD with learning rate $\propto \frac{1}{\sqrt{T}}$ for 500K gradient steps. We train Transformers for 80K gradient steps, with 10% label smoothing and no drop-out.

Label Noise. In our experiments, label noise of probability p refers to training on a samples which have the correct label with probability $(1-p)$, and a uniformly random incorrect label otherwise (label noise is sampled only once and not per epoch). Figure 1 plots test error on the noisy distribution, while the remaining figures plot test error with respect to the clean distribution (the two curves are just linear rescaling of one another).

5 Model-wise Double Descent



(a) **CIFAR-100.** There is a peak in test error even with no label noise.

(b) **CIFAR-10.** There is a "plateau" in test error around the interpolation point with no label noise, which develops into a peak for added label noise.

Figure 4: **Model-wise double descent for ResNet18s.** Trained on CIFAR-100 and CIFAR-10, with varying label noise. Optimized using Adam with LR 0.0001 for 4K epochs, and data-augmentation.

In this section, we study the test error of models of increasing size, when training to completion (for a fixed large number of optimization steps). We demonstrate model-wise double descent across different architectures, datasets, optimizers, and training procedures. The critical region exhibits distinctly different test behavior around the interpolation point and there is often a peak in test error that becomes more prominent in settings with label noise.

For the experiments in this section (Figures 4, 5, 6, 7, 8), notice that all modifications which increase the interpolation threshold (such as adding label noise, using data augmentation, and increasing the number of train samples) also correspondingly shift the peak in test error towards larger models. Additional plots showing the early-stopping behavior of these models, and additional experiments showing double descent in settings with no label noise (e.g. Figure 19) are in Appendix E.2. We also observed model-wise double descent for adversarial training, with a prominent robust test error peak even in settings without label noise. See Figure 26 in Appendix E.2.

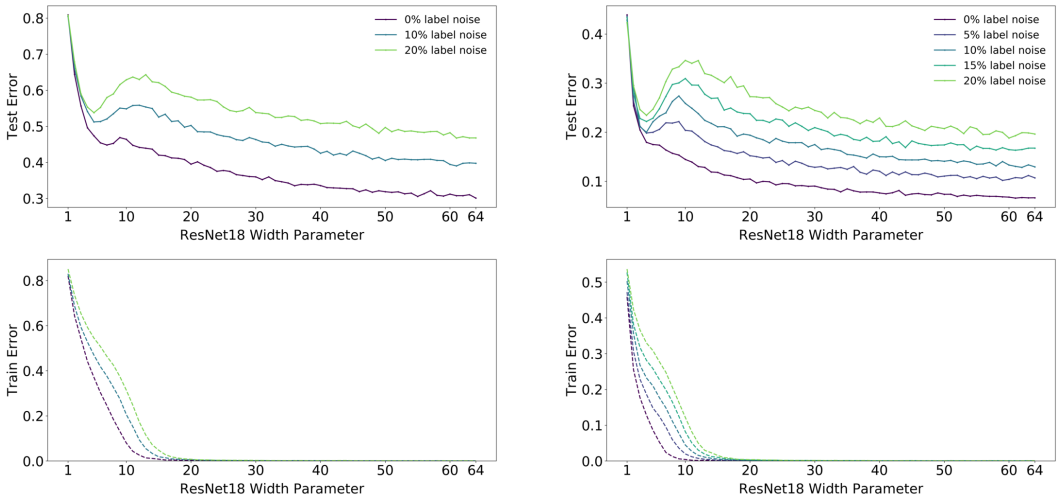
Discussion. Fully understanding the mechanisms behind model-wise double descent in deep neural networks remains an important open question. However, an analog of model-wise double descent occurs even for linear models. A recent stream of theoretical works analyzes this setting (Bartlett et al. (2019); Muthukumar et al. (2019); Belkin et al. (2019); Mei & Montanari (2019); Hastie et al. (2019)). We believe similar mechanisms may be at work in deep neural networks.

Informally, our intuition is that for model-sizes at the interpolation threshold, there is effectively only one model that fits the train data and this interpolating model is very sensitive to noise in the

对于 ResNets 和 CNNs，我们使用交叉熵损失进行训练，并使用以下优化器：（1）Adam，学习率为 0.0001，训练 4K 个周期；（2）SGD，学习率为 $\propto \frac{1}{\sqrt{t}}$，训练 500K 个梯度步骤。我们对 Transformers 进行 80K 个梯度步骤的训练，使用 10% 的标签平滑，并且不使用 drop out。

标签噪声。在我们的实验中，概率为 p 的标签噪声指的是在样本上进行训练，这些样本以概率 $(1 - p)$ 具有正确的标签，否则具有均匀随机的错误标签（标签噪声仅采样一次，而不是每个周期）。图1绘制了在噪声分布上的测试误差，而其余图形则绘制了相对于干净分布的测试误差（这两条曲线只是彼此的线性重新缩放）。

5 模型层面的双重下降



(a) CIFAR-100。即使没有标签噪声，测试错误率也会出现峰值。

(b) CIFAR-10。在没有标签噪声的插值点附近，测试误差出现了一个“平台”，而在添加标签噪声后，这个“平台”发展成了一个峰值。

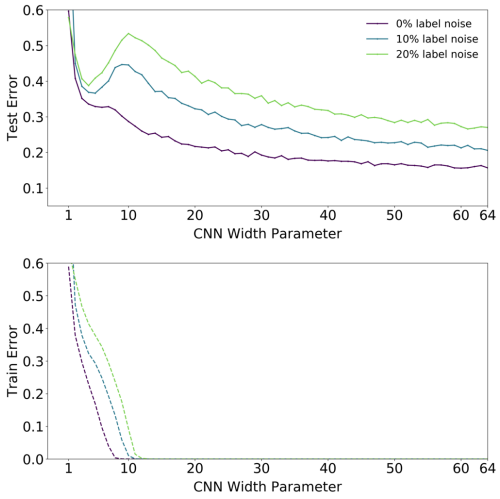
图 4：ResNet18s 的模型双重下降。在 CIFAR-100 和 CIFAR-10 上训练，标签噪声变化。使用 Adam 优化器，学习率为 0.0001，训练 4K 轮，并进行数据增强。

在本节中，我们研究在训练完成时（对于固定的大量优化步骤）模型大小增加时的测试误差。我们展示了在不同的架构、数据集、优化器和训练过程中的模型双重下降。关键区域在插值点附近表现出明显不同的测试行为，并且在有标签噪声的情况下，测试误差通常会出现更为显著的峰值。

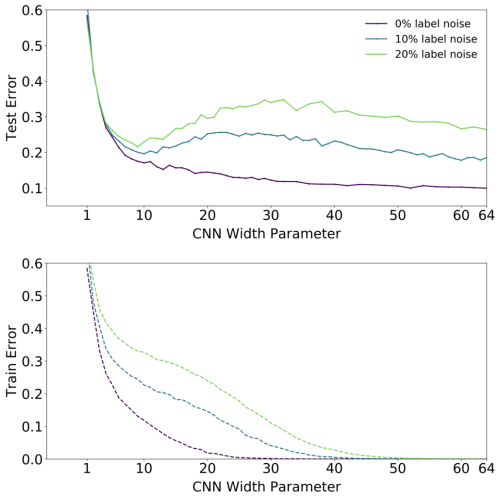
对于本节中的实验（图4、5、6、7、8），请注意，所有增加插值阈值的修改（例如添加标签噪声、使用数据增强和增加训练样本数量）也相应地将测试误差的峰值向更大的模型转移。附录E.2中有额外的图表显示这些模型的提前停止行为，以及在无标签噪声环境下显示双重下降的额外实验（例如图19）。我们还观察到对抗训练的模型双重下降，即使在没有标签噪声的环境中，也有明显的鲁棒测试误差峰值。详见附录E.2中的图26。

讨论。充分理解深度神经网络中模型双重下降背后的机制仍然是一个重要的未解之谜。然而，即使在线性模型中也会出现模型双重下降的类似现象。最近的一系列理论工作分析了这一情形（Bartlett 等人 (2019); Muthukumar 等人 (2019); Belkin 等人 (2019); Mei & Montanari (2019); Hastie 等人 (2019)）。我们相信类似的机制可能在深度神经网络中起作用。

非正式地说，我们的直觉是，对于处于插值阈值的模型大小，实际上只有一个模型适合训练数据，并且这个插值模型对噪声非常敏感。



(a) Without data augmentation.



(b) With data augmentation.

Figure 5: **Effect of Data Augmentation.** 5-layer CNNs on CIFAR10, with and without data-augmentation. Data-augmentation shifts the interpolation threshold to the right, shifting the test error peak accordingly. Optimized using SGD for 500K steps. See Figure 27 for larger models.

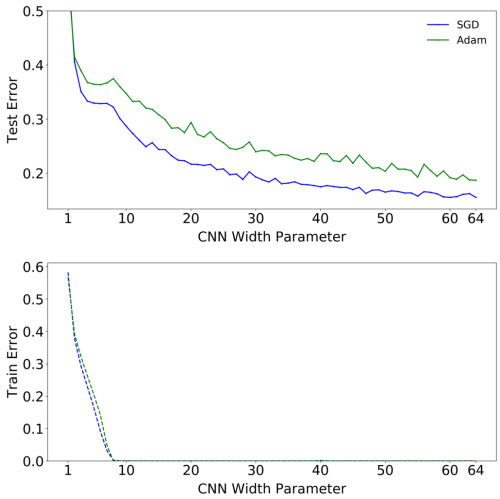


Figure 6: **SGD vs. Adam.** 5-Layer CNNs on CIFAR-10 with no label noise, and no data augmentation. Optimized using SGD for 500K gradient steps, and Adam for 4K epochs.

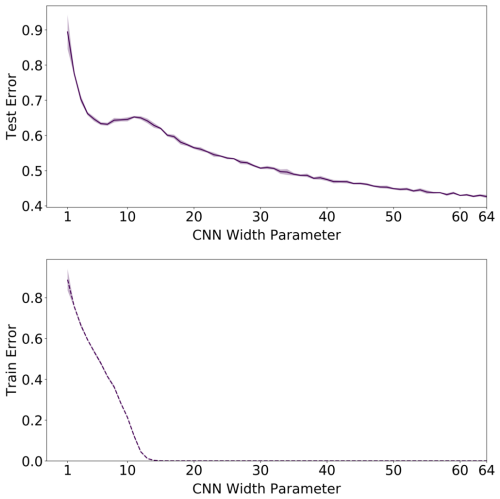
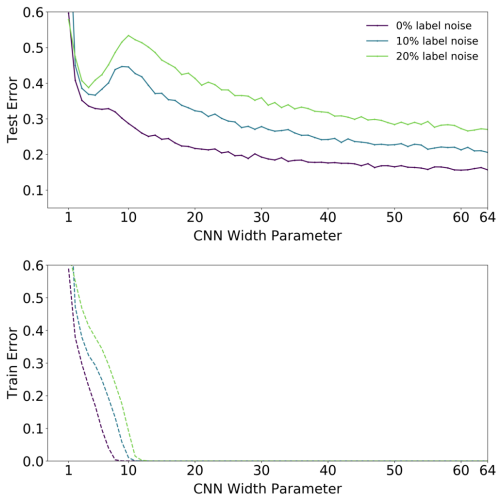


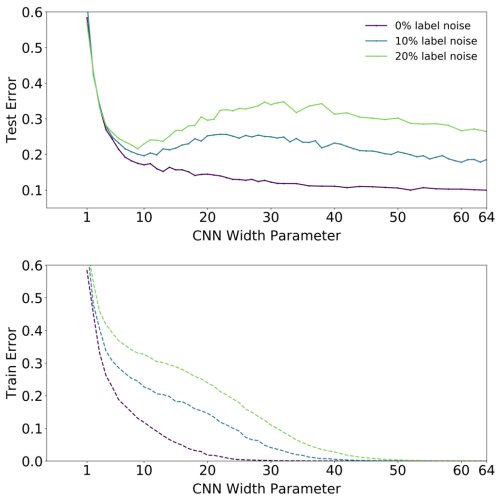
Figure 7: **Noiseless settings.** 5-layer CNNs on CIFAR-100 with no label noise; note the peak in test error. Trained with SGD and no data augmentation. See Figure 20 for the early-stopping behavior of these models.

train set and/or model mis-specification. That is, since the model is just barely able to fit the train data, forcing it to fit even slightly-noisy or mis-specified labels will destroy its global structure, and result in high test error. (See Figure 28 in the Appendix for an experiment demonstrating this noise sensitivity, by showing that ensembling helps significantly in the critically-parameterized regime). However for over-parameterized models, there are many interpolating models that fit the train set, and SGD is able to find one that "memorizes" (or "absorbs") the noise while still performing well on the distribution.

The above intuition is theoretically justified for linear models. In general, this situation manifests even without label noise for linear models (Mei & Montanari (2019)), and occurs whenever there



(a) Without data augmentation.



(b) With data augmentation.

图5：数据增强的效果。5层CNN在CIFAR10上的表现，分别在有和没有数据增强的情况下。数据增强将插值阈值向右移动，相应地移动测试误差峰值。使用SGD优化500K步。有关更大模型，请参见图27。

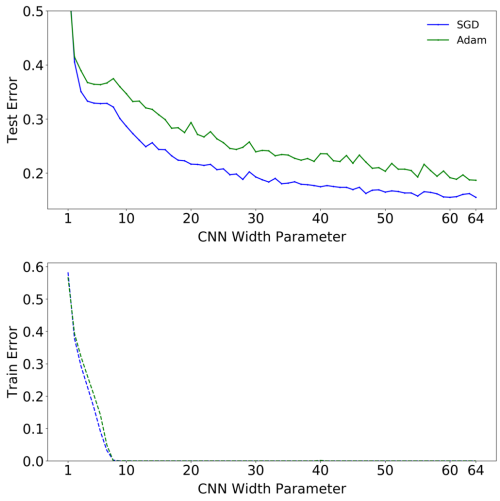


图6：SGD与Adam。5层CNN在CIFAR-10上，无标签噪声，无数据增强。使用SGD优化500K梯度步，使用Adam优化4K个epoch。

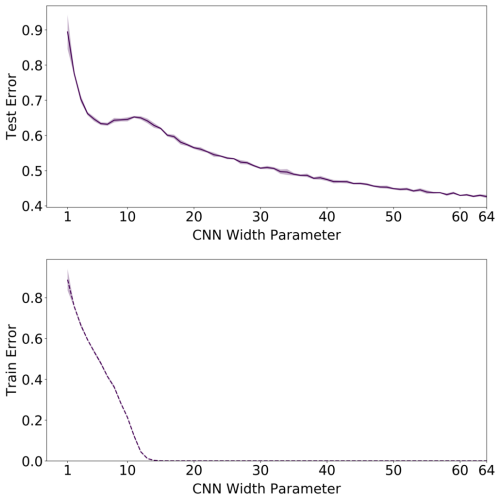


图7：无噪声设置。5层CNN在CIFAR-100上无标签噪声；注意测试误差的峰值。使用SGD训练且无数据增强。有关这些模型的早停行为，请参见图20。

训练集和/或模型错误指定。也就是说，由于模型仅能勉强拟合训练数据，强迫其拟合即使是稍微有噪声或错误指定的标签将破坏其整体结构，并导致高测试误差。（参见附录中的图28，通过展示集成在关键参数化状态下显著有助于减轻噪声敏感性来证明这一点的实验）。然而，对于过参数化模型，有许多插值模型可以拟合训练集，SGD能够找到一个“记住”（或“吸收”）噪声的模型，同时在分布上仍表现良好。

上述直觉在理论上对于线性模型是有依据的。一般来说，即使在线性模型中没有标签噪声，这种情况也会出现（Mei & Montanari (2019)），并且只要有这种情况发生。

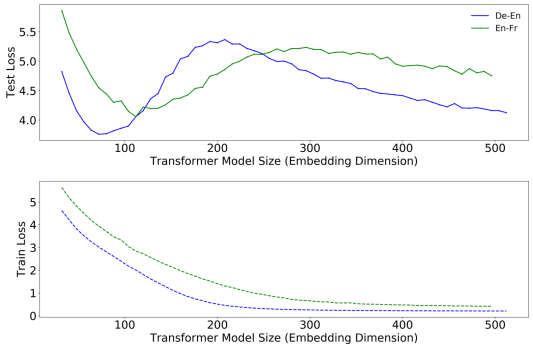


Figure 8: **Transformers on language translation tasks:** Multi-head-attention encoder-decoder Transformer model trained for 80k gradient steps with labeled smoothed cross-entropy loss on IWSLT'14 German-to-English (160K sentences) and WMT'14 English-to-French (subsampled to 200K sentences) dataset. Test loss is measured as per-token perplexity.

is *model mis-specification* between the structure of the true distribution and the model family. We believe this intuition extends to deep learning as well, and it is consistent with our experiments.

6 Epoch-wise Double Descent

In this section, we demonstrate a novel form of double-descent with respect to training epochs, which is consistent with our unified view of effective model complexity (EMC) and the generalized double descent hypothesis. Increasing the train time increases the EMC—and thus a sufficiently large model transitions from under- to over-parameterized over the course of training.

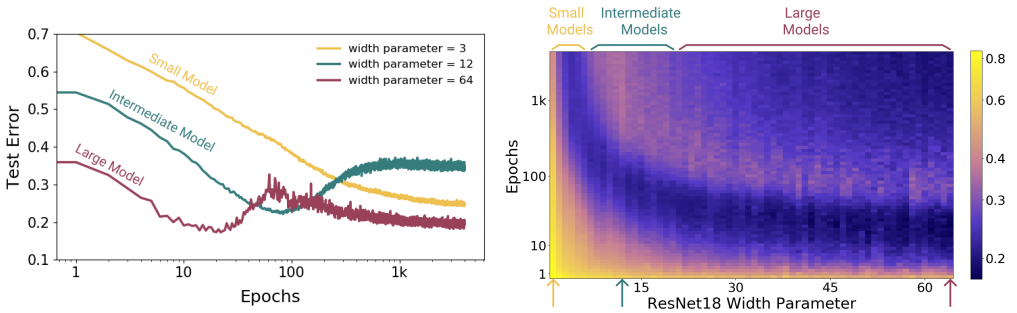


Figure 9: **Left:** Training dynamics for models in three regimes. Models are ResNet18s on CIFAR10 with 20% label noise, trained using Adam with learning rate 0.0001, and data augmentation. **Right:** Test error over (Model size $\times$ Epochs). Three slices of this plot are shown on the left.

As illustrated in Figure 9, sufficiently large models can undergo a "double descent" behavior where test error first decreases then increases near the interpolation threshold, and then decreases again. In contrast, for "medium sized" models, for which training to completion will only barely reach ≈ 0 error, the test error as a function of training time will follow a classical U-like curve where it is better to stop early. Models that are too small to reach the approximation threshold will remain in the "under parameterized" regime where increasing train time monotonically decreases test error. Our experiments (Figure 10) show that many settings of dataset and architecture exhibit epoch-wise double descent, in the presence of label noise. Further, this phenomenon is robust across optimizer variations and learning rate schedules (see additional experiments in Appendix E.1). As in model-wise double descent, the test error peak is accentuated with label noise.

Conventional wisdom suggests that training is split into two phases: (1) In the first phase, the network learns a function with a small generalization gap (2) In the second phase, the network starts to over-fit the data leading to an increase in test error. Our experiments suggest that this is not the complete picture—in some regimes, the test error decreases again and may achieve a lower value at the end of training as compared to the first minimum (see Fig 10 for 10% label noise).

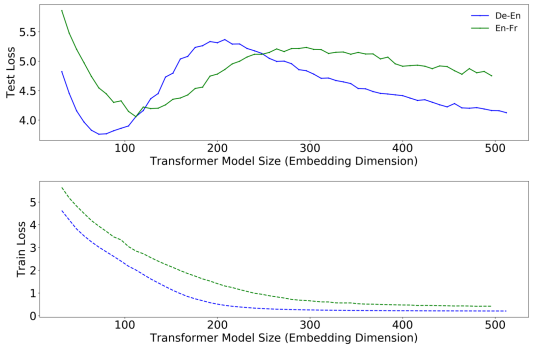


图 8：用于语言翻译任务的 Transformer ：多头注意力编码器-解码器 Transformer 模型在 IWSLT ‘ 14 德语到英语（160K 句子）和 WMT ‘ 14 英语到法语（抽样至 200K 句子）数据集上，经过 80k 梯度步骤训练，使用标签平滑交叉熵损失。测试损失以每个标记的困惑度衡量。

model mis-specification 是真实分布结构与模型族之间的。我们相信这种直觉也适用于深度学习，并且与我们的实验结果一致。

6 逐时代双重下降

在本节中，我们展示了一种关于训练周期的新型双重下降形式，这与我们对有效模型复杂性（EMC）和广义双重下降假设的统一观点是一致的。增加训练时间会增加EMC，因此一个足够大的模型在训练过程中会从欠参数化过渡到过参数化。

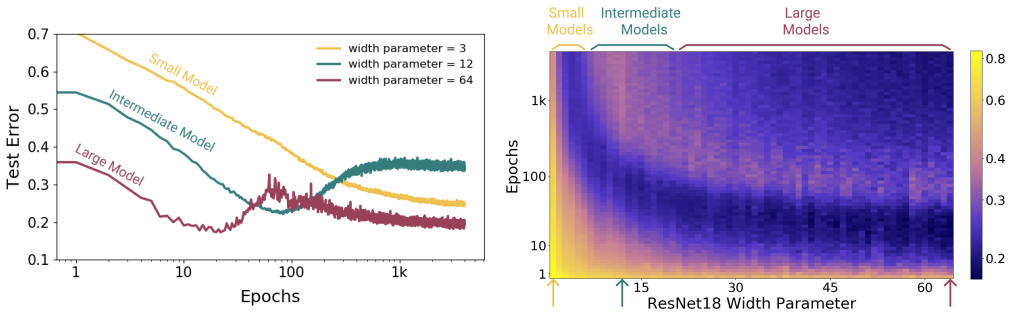
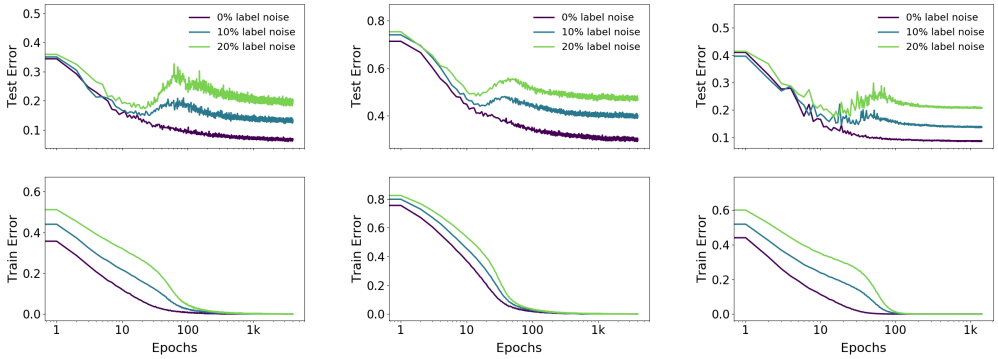


图9：左：三种状态下模型的训练动态。模型是使用20%标签噪声的CIFAR10上的ResNet18，使用Adam优化器，学习率为0.0001，并进行数据增强。右：测试误差随（模型大小 × 训练轮数）的变化。左侧显示了该图的三个切片。

如图9所示，足够大的模型可以经历 “ 双重下降 ” 行为，其中测试误差首先减少，然后在插值阈值附近增加，然后再次减少。相比之下，对于 “ 中等大小 ” 的模型，训练到完成时几乎只能达到≈ 0误差，测试误差作为训练时间的函数将遵循经典的U型曲线，因此最好提前停止。对于太小而无法达到近似阈值的模型，将保持在 “ 欠参数化 ” 状态，其中增加训练时间会单调地减少测试误差。我们的实验（图10）表明，许多数据集和架构设置在存在标签噪声的情况下表现出逐周期的双重下降。此外，这种现象在优化器变化和学习率计划中具有鲁棒性（参见附录E.1中的附加实验）。如同模型层面的双重下降，测试误差峰值在标签噪声的情况下更加明显。

传统观点认为训练分为两个阶段：（1）在第一阶段，网络学习一个具有小泛化间隙的函数（2）在第二阶段，网络开始对数据过拟合，导致测试误差增加。我们的实验表明，这并不是完整的情况——在某些情况下，测试误差再次减少，并可能在训练结束时达到比第一个最小值更低的值（参见图10，10%标签噪声）。



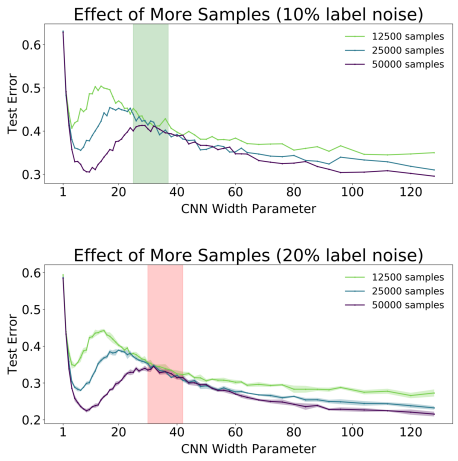
(a) ResNet18 on CIFAR10. (b) ResNet18 on CIFAR100. (c) 5-layer CNN on CIFAR 10.

Figure 10: **Epoch-wise double descent** for ResNet18 and CNN (width=128). ResNets trained using Adam with learning rate 0.0001, and CNNs trained with SGD with inverse-squareroot learning rate.

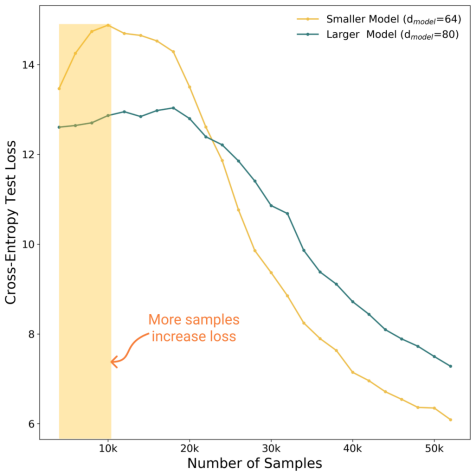
7 Sample-wise Non-monotonicity

In this section, we investigate the effect of varying the number of train samples, for a fixed model and training procedure. Previously, in model-wise and epoch-wise double descent, we explored behavior in the critical regime, where $\mathrm{EMC}_{\mathcal{D},\epsilon}(\mathcal{T}) \approx n$, by varying the EMC. Here, we explore the critical regime by varying the number of train samples n. By increasing n, the same training procedure $\mathcal{T}$ can switch from being effectively over-parameterized to effectively under-parameterized.

We show that increasing the number of samples has two different effects on the test error vs. model complexity graph. On the one hand, (as expected) increasing the number of samples shrinks the area under the curve. On the other hand, increasing the number of samples also has the effect of "shifting the curve to the right" and increasing the model complexity at which test error peaks.

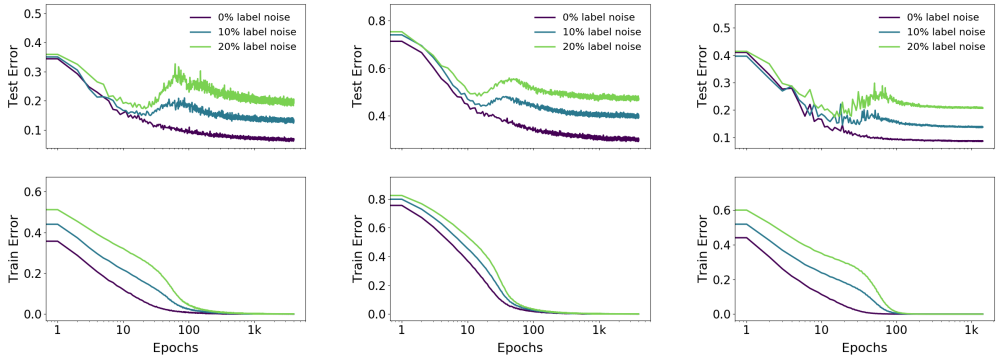


(a) Model-wise double descent for 5-layer CNNs on CIFAR-10, for varying dataset sizes. **Top:** There is a range of model sizes (shaded green) where training on $2\times$ more samples does not improve test error. **Bottom:** There is a range of model sizes (shaded red) where training on $4\times$ more samples does not improve test error.



(b) **Sample-wise non-monotonicity.** Test loss (per-word perplexity) as a function of number of train samples, for two transformer models trained to completion on IWSLT'14. For both model sizes, there is a regime where more samples hurt performance. Compare to Figure 3, of model-wise double-descent in the identical setting.

Figure 11: Sample-wise non-monotonicity.



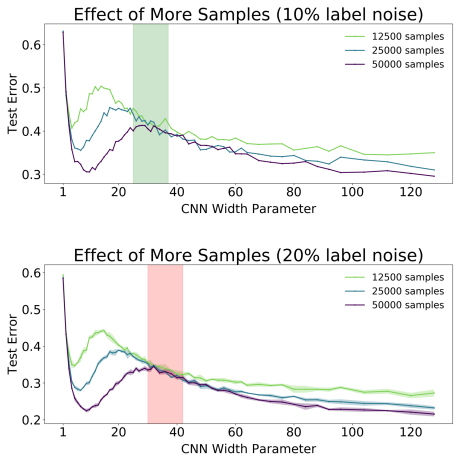
(a) ResNet18 on CIFAR10. (b) ResNet18 on CIFAR100. (c) 5-layer CNN on CIFAR 10.

图 10：ResNet18 和 CNN（宽度=128）的逐周期双重下降。使用 Adam 以学习率 0.0001 训练 ResNet，使用 SGD 以反平方根学习率训练 CNN。

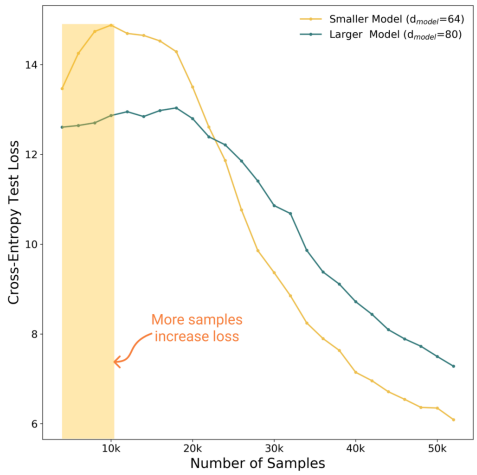
7 样本非单调性

在本节中，我们研究在固定模型和训练过程的情况下，改变训练样本数量的影响。之前，在模型层面和周期层面的双重下降中，我们通过改变EMC来探索$\text{EMC}_{\mathcal{D},\epsilon}(\mathcal{T}) \approx n$的临界状态。在这里，我们通过改变训练样本数量$n$来探索临界状态。通过增加$n$，相同的训练过程$\mathcal{T}$可以从有效的过参数化转变为有效的欠参数化。

我们表明，增加样本数量对测试误差与模型复杂性图有两种不同的影响。一方面，（如预期的那样）增加样本数量会缩小曲线下的面积。另一方面，增加样本数量也会导致“曲线向右移动”，并增加测试误差达到峰值时的模型复杂性。



(a) 针对 CIFAR-10 上的 5 层 CNN 的模型双重下降，数据集大小变化。顶部：存在一系列模型大小（绿色阴影），在此范围内训练 2× 更多样本不会改善测试误差。底部：存在一系列模型大小（红色阴影），在此范围内训练 4× 更多样本不会改善测试误差。



(b) 样本非单调性。测试损失（每词困惑度）作为训练样本数量的函数，针对两个在 IWSLT ’14 上训练至完成的 transformer 模型。对于这两种模型大小，存在一个更多样本会损害性能的情况。与图 3 中在相同设置下的模型双重下降进行比较。

图 11：样本非单调性。

These twin effects are shown in Figure 11a. Note that there is a range of model sizes where the effects "cancel out"—and having $4\times$ more train samples does not help test performance when training to completion. Outside the critically-parameterized regime, for sufficiently under- or over-parameterized models, having more samples helps. This phenomenon is corroborated in Figure 12, which shows test error as a function of both model and sample size, in the same setting as Figure 11a.

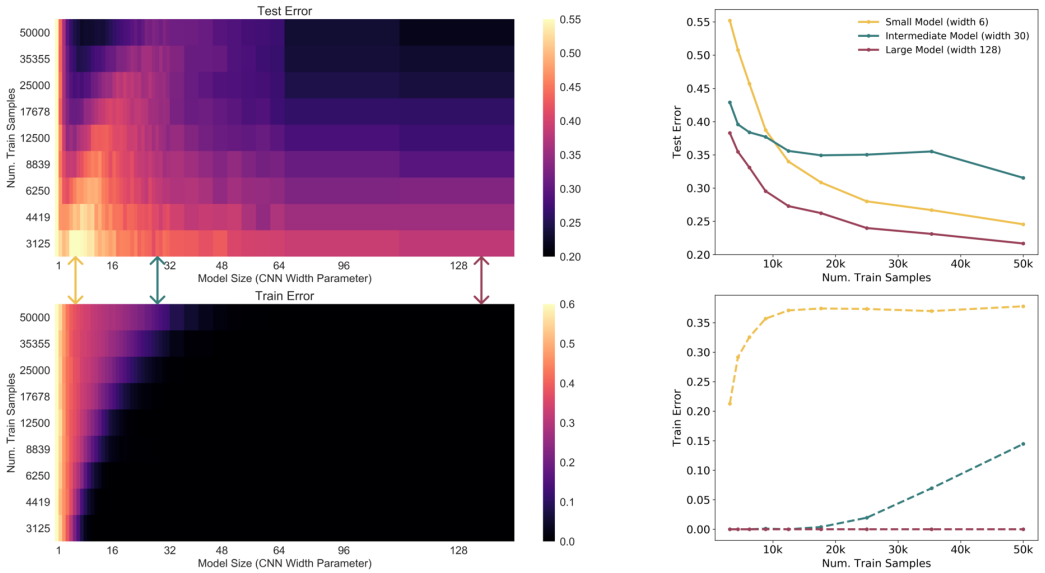


Figure 12: **Left:** Test Error as a function of model size and number of train samples, for 5-layer CNNs on CIFAR-10 + 20% noise. Note the ridge of high test error again lies along the interpolation threshold. **Right:** Three slices of the left plot, showing the effect of more data for models of different sizes. Note that, when training to completion, more data helps for small and large models, but does not help for near-critically-parameterized models (green).

In some settings, these two effects combine to yield a regime of model sizes where more data actually hurts test performance as in Figure 3 (see also Figure 11b). Note that this phenomenon is not unique to DNNs: more data can hurt even for linear models (see Appendix D).

8 Conclusion and Discussion

We introduce a generalized double descent hypothesis: models and training procedures exhibit atypical behavior when their Effective Model Complexity is comparable to the number of train samples. We provide extensive evidence for our hypothesis in modern deep learning settings, and show that it is robust to choices of dataset, architecture, and training procedures. In particular, we demonstrate "model-wise double descent" for modern deep networks and characterize the regime where bigger models can perform worse. We also demonstrate "epoch-wise double descent," which, to the best of our knowledge, has not been previously proposed. Finally, we show that the double descent phenomenon can lead to a regime where training on more data leads to worse test performance. Preliminary results suggest that double descent also holds as we vary the amount of regularization for a fixed model (see Figure 22).

We also believe our characterization of the critical regime provides a useful way of thinking for practitioners—if a model and training procedure are just barely able to fit the train set, then small changes to the model or training procedure may yield unexpected behavior (e.g. making the model slightly larger or smaller, changing regularization, etc. may hurt test performance).

Early stopping. We note that many of the phenomena that we highlight often do not occur with optimal early-stopping. However, this is consistent with our generalized double descent hypothesis: if early stopping prevents models from reaching 0 train error then we would not expect to see double-descent, since the EMC does not reach the number of train samples. Further, we show at least one

这两个双重效应如图11a所示。请注意，在某些模型规模范围内，这些效应会“ 相互抵消 ”——当训练到完成时，拥有4×更多的训练样本并不会提高测试性能。在临界参数化范围之外，对于足够欠参数化或过参数化的模型，拥有更多样本是有帮助的。图12证实了这一现象，该图显示了在与图11a相同的设置下，测试误差作为模型和样本大小的函数的变化。

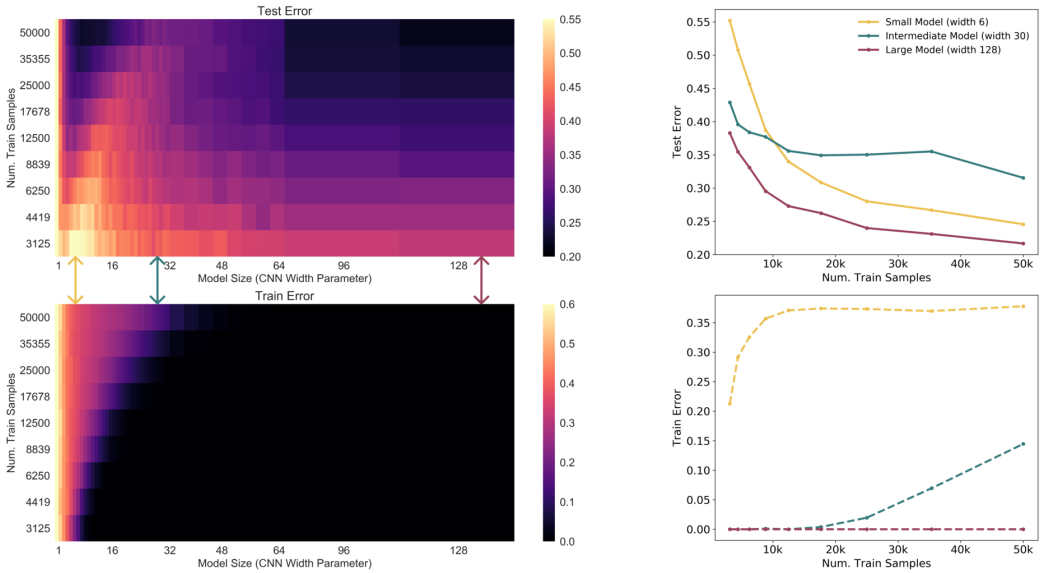


图12：左图：测试误差作为模型大小和训练样本数量的函数，针对CIFAR-10 + 20%噪声的5层CNN。注意高测试误差的脊线再次沿着插值阈值。右图：左图的三个切片，显示了更多数据对不同大小模型的影响。注意，当训练完成时，更多的数据对小型和大型模型有帮助，但对接近临界参数化的模型（绿色）没有帮助。

在某些情况下，这两种效应结合在一起，导致模型大小的一个范围，在这个范围内，更多的数据实际上会损害测试性能，如图3所示（另见图11b）。请注意，这种现象并非DNNs独有：即使对于线性模型，更多的数据也可能有害（见附录D）。

8 结论与讨论

我们引入了一个广义的双重下降假设：当模型的有效模型复杂度与训练样本数量相当时，模型和训练过程会表现出非典型行为。我们在现代深度学习环境中为我们的假设提供了广泛的证据，并表明它对数据集、架构和训练过程的选择具有鲁棒性。特别是，我们展示了现代深度网络的“ 模型双重下降 ”，并描述了更大模型可能表现更差的情况。我们还展示了“ 历元双重下降 ”，据我们所知，这在以前没有被提出过。最后，我们表明双重下降现象可能导致一种情况，即在更多数据上进行训练会导致更差的测试性能。初步结果表明，当我们对固定模型调整正则化量时，双重下降也成立（见图22）。

我们也认为，我们对关键状态的描述为从业者提供了一种有用的思考方式——如果一个模型和训练过程仅仅能够勉强拟合训练集，那么对模型或训练过程的微小改变可能会产生意想不到的行为（例如，使模型稍微大一点或小一点，改变正则化等可能会损害测试性能）。

提前停止。我们注意到，我们强调的许多现象通常不会在最佳提前停止的情况下发生。然而，这与我们的广义双重下降假设是一致的：如果提前停止阻止模型达到0训练误差，那么我们不期望看到双重下降，因为EMC没有达到训练样本的数量。此外，我们至少展示了一个

setting where model-wise double descent can still occur even with optimal early stopping (ResNets on CIFAR-100 with no label noise, see Figure 19). We have not observed settings where more data hurts when optimal early-stopping is used. However, we are not aware of reasons which preclude this from occurring. We leave fully understanding the optimal early stopping behavior of double descent as an important open question for future work.

Label Noise. In our experiments, we observe double descent most strongly in settings with label noise. However, we believe this effect is not fundamentally about label noise, but rather about *model mis-specification*. For example, consider a setting where the label noise is not truly random, but rather pseudorandom (with respect to the family of classifiers being trained). In this setting, the performance of the Bayes optimal classifier would not change (since the pseudorandom noise is deterministic, and invertible), but we would observe an identical double descent as with truly random label noise. Thus, we view adding label noise as merely a proxy for making distributions "harder"— i.e. increasing the amount of model mis-specification.

Other Notions of Model Complexity. Our notion of *Effective Model Complexity* is related to classical complexity notions such as Rademacher complexity, but differs in several crucial ways: (1) EMC depends on the *true labels* of the data distribution, and (2) EMC depends on the training procedure, not just the model architecture.

Other notions of model complexity which do not incorporate features (1) and (2) would not suffice to characterize the location of the double-descent peak. Rademacher complexity, for example, is determined by the ability of a model architecture to fit a randomly-labeled train set. But Rademacher complexity and VC dimension are both insufficient to determine the model-wise double descent peak location, since they do not depend on the distribution of labels— and our experiments show that adding label noise shifts the location of the peak.

Moreover, both Rademacher complexity and VC dimension depend only on the model family and data distribution, and not on the training procedure used to find models. Thus, they are not capable of capturing train-time double-descent effects, such as "epoch-wise" double descent, and the effect of data-augmentation on the peak location.

ACKNOWLEDGMENTS

We thank Mikhail Belkin for extremely useful discussions in the early stages of this work. We thank Christopher Olah for suggesting the Model Size $\times$ Epoch visualization, which led to the investigation of epoch-wise double descent, as well as for useful discussion and feedback. We also thank Alec Radford, Jacob Steinhardt, and Vaishaal Shankar for helpful discussion and suggestions. P.N. thanks OpenAI, the Simons Institute, and the Harvard Theory Group for a research environment that enabled this kind of work.

We thank Dimitris Kalimeris, Benjamin L. Edelman, and Sharon Qian, and Aditya Ramesh for comments on an early draft of this work.

This work supported in part by NSF grant CAREER CCF 1452961, BSF grant 2014389, NSF US-ICCS proposal 1540428, a Google Research award, a Facebook research award, a Simons Investigator Award, a Simons Investigator Fellowship, and NSF Awards CCF 1715187, CCF 1565264, CCF 1301976, IIS 1409097, and CNS 1618026. Y.B. would like to thank the MIT-IBM Watson AI Lab for contributing computational resources for experiments.

REFERENCES

Madhu S Advani and Andrew M Saxe. High-dimensional dynamics of generalization error in neural networks. *arXiv preprint arXiv:1710.03667*, 2017.

Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *arXiv preprint arXiv:1906.11300*, 2019.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine learning and the bias-variance trade-off. *arXiv preprint arXiv:1812.11118*, 2018.

即使在最佳提前停止的情况下，模型双重下降仍可能发生的设置（CIFAR-100上的ResNets，无标签噪声，见图19）。我们尚未观察到在使用最佳提前停止时更多数据会有害的情况。然而，我们也不知道有什么理由可以排除这种情况的发生。我们将完全理解双重下降的最佳提前停止行为作为未来研究的一个重要开放问题。

标签噪声。在我们的实验中，我们观察到双重下降在有标签噪声的设置中最为明显。然而，我们认为这种效果并不是根本上关于标签噪声，而是关于*model mis-specification*。例如，考虑一种情况，其中标签噪声并不是真正随机的，而是伪随机的（相对于正在训练的分类器家族）。在这种情况下，贝叶斯最优分类器的性能不会改变（因为伪随机噪声是确定性的且可逆的），但我们会观察到与真正随机标签噪声相同的双重下降。因此，我们认为添加标签噪声仅仅是使分布“更难”的一种代理——即增加模型错误指定的程度。

模型复杂性的其他概念。我们对*Effective Model Complexity*的概念与经典的复杂性概念（如Rademacher复杂性）有关，但在几个关键方面有所不同：(1) EMC依赖于数据分布的*true labels*，(2) EMC依赖于训练过程，而不仅仅是模型架构。

其他不包含特征 (1) 和 (2) 的模型复杂性概念不足以描述双下降峰的位置。例如，Rademacher 复杂性由模型架构拟合随机标记的训练集的能力决定。但是，Rademacher 复杂性和 VC 维度都不足以确定模型的双下降峰位置，因为它们不依赖于标签的分布——我们的实验表明，添加标签噪声会改变峰的位置。

此外，Rademacher复杂度和VC维数仅依赖于模型家族和数据分布，而不依赖于用于寻找模型的训练过程。因此，它们无法捕捉训练时的双重下降效应，例如“逐轮”双重下降，以及数据增强对峰值位置的影响。

致谢

我们感谢 Mikhail Belkin 在这项工作早期阶段提供的极其有用的讨论。我们感谢 Christopher Olah 提出模型大小 × 时代可视化的建议，这导致了对逐时代双重下降的研究，并感谢他提供的有用讨论和反馈。我们还感谢 Alec Radford、Jacob Steinhardt 和 Vaishaal Shankar 的有益讨论和建议。P.N. 感谢 OpenAI、西蒙斯研究所和哈佛理论小组提供的研究环境，使这类工作成为可能。

我们感谢 Dimitris Kalimeris、Benjamin L. Edelman、Sharon Qian 和 Aditya Ramesh 对本工作早期草稿的评论。

本工作部分由NSF资助项目CAREER CCF 1452961、BSF资助项目2014389、NSF US-ICCS提案1540428、谷歌研究奖、Facebook研究奖、Simons Investigator奖、Simons Investigator奖学金以及NSF奖项CCF 1715187、CCF 1565264、CCF 1301976、IIS 1409097和CNS 1618026支持。Y.B.感谢MIT-IBM Watson AI实验室为实验提供计算资源。

参考文献

Madhu S Advani 和 Andrew M Saxe. 神经网络中泛化误差的高维动态。*arXiv preprint arXiv:1710.03667*, 2017.

Peter L Bartlett, Philip M Long, Gábor Lugosi, 和 Alexander Tsigler。线性回归中的良性过拟合。*arXiv preprint arXiv:1906.11300*, 2019.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, 和 Soumik Mandal。调和现代机器学习与偏差-方差权衡。*arXiv preprint arXiv:1812.11118*, 2018.

Mikhail Belkin, Daniel Hsu, and Ji Xu. Two models of double descent for weak features. *arXiv preprint arXiv:1903.07571*, 2019.

Koby Bibas, Yaniv Fogel, and Meir Feder. A new look at an old problem: A universal learning approach to linear regression. *arXiv preprint arXiv:1905.04708*, 2019.

Mauro Cettolo, Christian Girardi, and Marcello Federico. Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pp. 261–268, Trento, Italy, May 2012.

Mario Geiger, Arthur Jacot, Stefano Spigler, Franck Gabriel, Levent Sagun, Stéphane d'Ascoli, Giulio Biroli, Clément Hongler, and Matthieu Wyart. Scaling description of generalization with number of parameters in deep learning. *arXiv preprint arXiv:1901.01608*, 2019a.

Mario Geiger, Stefano Spigler, Stéphane d'Ascoli, Levent Sagun, Marco Baity-Jesi, Giulio Biroli, and Matthieu Wyart. Jamming transition as a paradigm to understand the loss landscape of deep neural networks. *Physical Review E*, 100(1):012115, 2019b.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pp. 630–645. Springer, 2016.

Yanping Huang, Yonglong Cheng, Dehao Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V. Le, and Zhifeng Chen. Gpipe: Efficient training of giant neural networks using pipeline parallelism. *CoRR*, abs/1811.06965, 2018. URL `http://arxiv.org/abs/1811.06965`.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

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

Partha P. Mitra. Understanding overfitting peaks in generalization error: Analytical risk curves for l2 and l1 penalized interpolation. *ArXiv*, abs/1906.03667, 2019.

Vidya Muthukumar, Kailas Vodrahalli, and Anant Sahai. Harmless interpolation of noisy data in regression. *arXiv preprint arXiv:1903.09139*, 2019.

Preetum Nakkiran, Gal Kaplun, Dimitris Kalimeris, Tristan Yang, Benjamin L Edelman, Fred Zhang, and Boaz Barak. Sgd on neural networks learns functions of increasing complexity. *arXiv preprint arXiv:1905.11604*, 2019.

Manfred Opper. Statistical mechanics of learning: Generalization. *The Handbook of Brain Theory and Neural Networks, 922-925.*, 1995.

Manfred Opper. Learning to generalize. *Frontiers of Life, 3(part 2), pp.763-775.*, 2001.

Mikhail Belkin, Daniel Hsu, 和 Ji Xu。弱特征的双重下降的两种模型。*arXiv preprint arXiv:1903.07571*, 2019.

Koby Bibas, Yaniv Fogel 和 Meir Feder. 以新视角看待老问题：线性回归的通用学习方法。*arXiv preprint arXiv:1905.04708*, 2019.

Mauro Cettolo, Christian Girardi, 和 Marcello Federico。Wit^3：转录和翻译演讲的网络库存。在 *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*，第 261 – 268 页，意大利特伦托，2012 年 5 月。

Mario Geiger, Arthur Jacot, Stefano Spigler, Franck Gabriel, Levent Sagun, Stéphane d ' Ascoli, Giulio Biroli, Clément Hongler, 和 Matthieu Wyart。深度学习中参数数量与泛化的缩放描述。*arXiv preprint arXiv:1901.01608*, 2019a.

Mario Geiger、Stefano Spigler、Stéphane d ' Ascoli、Levent Sagun、Marco Baity-Jesi、Giulio Biroli 和 Matthieu Wyart。堵塞转变作为理解深度神经网络损失景观的范例。*Physical Review E*, 100(1):012115, 2019b.

Ian J Goodfellow, Jonathon Shlens 和 Christian Szegedy. 解释和利用对抗性样本。*arXiv preprint arXiv:1412.6572*, 2014。Trevor Hastie, Robert Tibshirani, Jerome Friedman 和 James Franklin. 统计学习的要素：数据挖掘、推断和预测。*The Mathematical Intelligencer*, 27(2):83 – 85, 2005。Trevor Hastie, Andrea Montanari, Saharon Rosset 和 Ryan J Tibshirani. 高维无岭最小二乘插值中的惊喜。*arXiv preprint arXiv:1903.08560*, 2019。Kaiming He, Xiangyu Zhang, Shaoqing Ren 和 Jian Sun. 深度残差网络中的身份映射。在 *European conference on computer vision*, pp. 630 – 645. Springer, 2016。Yanping Huang, Yonglong Cheng, Dehao Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V. Le 和 Zhifeng Chen. Gpipe: 使用流水线并行训练巨型神经网络的高效方法。*CoRR*, abs/1811.06965, 2018。URL http://arxiv.org/abs/1811.06965。Alex Krizhevsky. 从微小图像中学习多层特征。技术报告, 2009。Alex Krizhevsky, Ilya Sutskever 和 Geoffrey E Hinton. 使用深度卷积神经网络进行Imagenet分类。在 *Advances in neural information processing systems*, pp. 1097 – 1105, 2012。Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras 和 Adrian Vladu. 迈向对抗攻击抵抗的深度学习模型。*arXiv preprint arXiv:1706.06083*, 2017。Song Mei 和 Andrea Montanari. 随机特征回归的泛化误差：精确渐近和双降曲线。*arXiv preprint arXiv:1908.05355*, 2019。

Partha P. Mitra. 理解泛化误差中的过拟合峰值：l2 和 l1 惩罚插值的解析风险曲线。*ArXiv*, abs/1906.03667, 2019.

Vidya Muthukumar, Kailas Vodrahalli, 和 Anant Sahai. 回归中对噪声数据的无害插值。*arXiv preprint arXiv:1903.09139*, 2019.

Preetum Nakkiran, Gal Kaplun, Dimitris Kalimeris, Tristan Yang, Benjamin L Edelman, Fred Zhang, 和 Boaz Barak。神经网络上的SGD学习复杂性递增的函数。*arXiv preprint arXiv:1905.11604*, 2019.

Manfred Opper. 学习的统计力学：泛化。*The Handbook of Brain Theory and Neural Networks, 922-925.*, 1995. Manfred Opper. 学习泛化。*Frontiers of Life, 3(part 2), pp.763-775.*, 2001.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

David Page. How to train your resnet. https://myrtle.ai/how-to-train-your-resnet-4-architecture/, 2018.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NeurIPS Autodiff Workshop*, 2017.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pp. 1177–1184, 2008.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *ArXiv*, abs/1508.07909, 2015.

Stefano Spigler, Mario Geiger, Stéphane d'Ascoli, Levent Sagun, Giulio Biroli, and Matthieu Wyart. A jamming transition from under-to over-parametrization affects loss landscape and generalization. *arXiv preprint arXiv:1810.09665*, 2018.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, 2015. URL http://arxiv.org/abs/1409.4842.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *ICLR*, abs/1611.03530, 2016.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, 和 Michael Auli。fairseq：一个快速、可扩展的序列建模工具包。在 *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019。

大卫·佩奇。如何训练你的resnet。https://myrtle.ai/how-to-train-your-resnet-4-architecture/, 2018。

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, 和 Adam Lerer。PyTorch中的自动微分。在 *NeurIPS Autodiff Workshop*, 2017。

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei 和 Ilya Sutskever。语言模型是无监督的多任务学习者。2019年。

阿里·拉希米和本杰明·雷希特。大规模核机器的随机特征。在 *Advances in neural information processing systems*，第 1177 – 1184 页，2008 年。

Rico Sennrich, Barry Haddow 和 Alexandra Birch。使用子词单元进行稀有词的神经机器翻译。*ArXiv*, abs/1508.07909, 2015。

Stefano Spigler, Mario Geiger, Stéphane d’Ascoli, Levent Sagun, Giulio Biroli, 和 Matthieu Wyart。一个从欠参数化到过参数化的堵塞转变影响损失景观和泛化。*arXiv preprint arXiv:1810.09665*, 2018.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, 和 Andrew Rabinovich。通过卷积深入研究。在 *Computer Vision and Pattern Recognition (CVPR)*, 2015。URL http://arxiv.org/abs/1409.4842。

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, 和 Illia Polosukhin。注意力机制是你所需要的一切。*CoRR*, abs/1706.03762, 2017。

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, 和 Oriol Vinyals。理解深度学习需要重新思考泛化。*ICLR*, abs/1611.03530, 2016。

A SUMMARY TABLE OF EXPERIMENTAL RESULTS

					Double-Descent		
Dataset	Architecture	Opt.	Aug.	% Noise	Model	Epoch	Figure(s)
CIFAR 10	CNN	SGD	✓	0	✗	✗	5, 27
			✓	10	✓	✓	5, 27, 6
			✓	20	✓	✓	5, 27
				0	✗	✗	5, 25
				10	✓	✓	5
				20	✓	✓	5
		SGD + w.d.	✓	20	✓	✓	21
		Adam		0	✓	–	25
	ResNet	Adam	✓	0	✗	✗	4, 10
			✓	5	✓	–	4
			✓	10	✓	✓	4, 10
			✓	15	✓	✓	4, 2
			✓	20	✓	✓	4, 9, 10
		Various	✓	20	–	✓	16, 17, 18
(subsampled)	CNN	SGD	✓	10	✓	–	11a
		SGD	✓	20	✓	–	11a, 12
(adversarial)	ResNet	SGD		0	Robust err.	–	26
CIFAR 100	ResNet	Adam	✓	0	✓	✗	4, 19, 10
			✓	10	✓	✓	4, 10
			✓	20	✓	✓	4, 10
	CNN	SGD		0	✓	✗	20
IWSLT '14 de-en	Transformer	Adam		0	✓	✗	8, 24
(subsampled)	Transformer	Adam		0	✓	✗	11b, 23
WMT '14 en-fr	Transformer	Adam		0	✓	✗	8, 24

B APPENDIX: EXPERIMENTAL DETAILS

B.1 MODELS

We use the following families of architectures. The PyTorch Paszke et al. (2017) specification of our ResNets and CNNs are available at `https://gitlab.com/harvard-machine-learning/double-descent/tree/master`.

ResNets. We define a family of ResNet18s of increasing size as follows. We follow the Preactivation ResNet18 architecture of He et al. (2016), using 4 ResNet blocks, each consisting of two BatchNorm-ReLU-Convolution layers. The layer widths for the 4 blocks are $[k, 2k, 4k, 8k]$ for varying $k \in \mathbb{N}$ and the strides are [1, 2, 2, 2]. The standard ResNet18 corresponds to $k = 64$ convolutional channels in the first layer. The scaling of model size with k is shown in Figure 13b. Our implementation is adapted from `https://github.com/kuangliu/pytorch-cifar`.

Standard CNNs. We consider a simple family of 5-layer CNNs, with four Conv-BatchNorm-ReLU-MaxPool layers and a fully-connected output layer. We scale the four convolutional layer widths as $[k, 2k, 4k, 8k]$. The MaxPool is [1, 2, 2, 8]. For all the convolution layers, the kernel size = 3, stride = 1 and padding=1. This architecture is based on the "backbone" architecture from Page (2018). For $k = 64$, this CNN has 1558026 parameters and can reach $> 90\%$ test accuracy on CIFAR-10 (Krizhevsky (2009)) with data-augmentation. The scaling of model size with k is shown in Figure 13a.

Transformers. We consider the encoder-decoder Transformer model from Vaswani et al. (2017) with 6 layers and 8 attention heads per layer, as implemented by fairseq Ott et al. (2019). We scale the size of the network by modifying the embedding dimension (d_{model}), and scale the width of the fully-connected layers proportionally ($d_{\text{ff}} = 4d_{\text{model}}$). We train with 10% label smoothing and no drop-out, for 80 gradient steps.

A SUMMARY TABLE OF EXPERIMENTAL RESULTS

					Double-Descent		
Dataset	Architecture	Opt.	Aug.	% Noise	Model	Epoch	Figure(s)
CIFAR 10	CNN	SGD	✓	0	✗	✗	5, 27
			✓	10	✓	✓	5, 27, 6
			✓	20	✓	✓	5, 27
				0	✗	✗	5, 25
				10	✓	✓	5
				20	✓	✓	5
		SGD + w.d.	✓	20	✓	✓	21
		Adam		0	✓	–	25
	ResNet	Adam	✓	0	✗	✗	4, 10
			✓	5	✓	–	4
			✓	10	✓	✓	4, 10
			✓	15	✓	✓	4, 2
			✓	20	✓	✓	4, 9, 10
		Various	✓	20	–	✓	16, 17, 18
(subsampled)	CNN	SGD	✓	10	✓	–	11a
		SGD	✓	20	✓	–	11a, 12
(adversarial)	ResNet	SGD		0	Robust err.	–	26
CIFAR 100	ResNet	Adam	✓	0	✓	✗	4, 19, 10
			✓	10	✓	✓	4, 10
			✓	20	✓	✓	4, 10
	CNN	SGD		0	✓	✗	20
IWSLT '14 de-en	Transformer	Adam		0	✓	✗	8, 24
(subsampled)	Transformer	Adam		0	✓	✗	11b, 23
WMT '14 en-fr	Transformer	Adam		0	✓	✗	8, 24

B 附录：实验细节

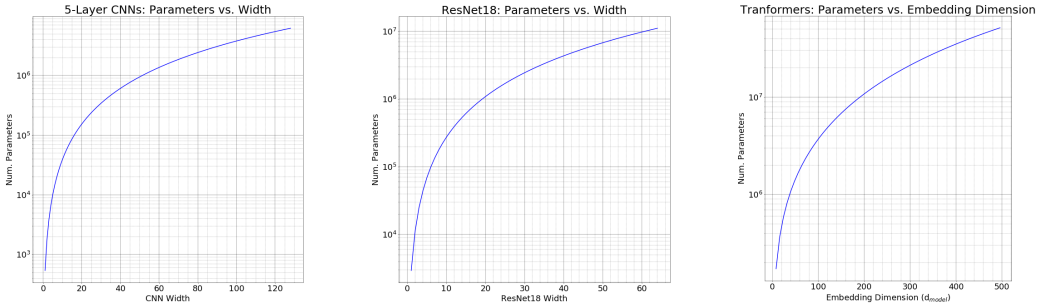
B.1 模型

我们使用以下架构系列。我们的 ResNets 和 CNNs 的 PyTorch Paszke 等人 (2017) 规范可在 https://gitlab.com/harvard-machine-learning/double-descent/tree/master 获取。

ResNets。我们定义了一系列不断增大的ResNet18，如下所示。我们遵循He等人（2016）的预激活ResNet18架构，使用4个ResNet块，每个块由两个BatchNorm-ReLU-Convolution层组成。对于不同的$k \in \mathbb{N}$，4个块的层宽为[k, $2k$, $4k$, $8k$]，步幅为[1, 2, 2, 2]。标准的ResNet18对应于第一层中的$k = 64$个卷积通道。模型大小随k的缩放如图13b所示。我们的实现改编自https://github.com/kuangliu/pytorch-cifar。

标准CNN。我们考虑一个简单的5层CNN家族，包含四个Conv-BatchNorm-ReLU-MaxPool层和一个全连接输出层。我们将四个卷积层的宽度按比例缩放为[k, $2k$, $4k$, $8k$]。MaxPool为[1, 2, 2, 8]。对于所有卷积层，核大小为= 3，步幅为= 1，填充为= 1。该架构基于Page (2018)的“骨干”架构。对于$k = 64$，这个CNN有1558026个参数，并且在CIFAR-10 (Krizhevsky (2009))上通过数据增强可以达到$>$ 90%的测试准确率。模型大小随k的缩放如图13a所示。

Transformer。我们考虑Vaswani等人（2017）提出的编码器-解码器Transformer模型，该模型具有6层，每层有8个注意力头，由fairseq Ott等人（2019）实现。我们通过修改嵌入维度（d_{model}）来调整网络的大小，并按比例调整全连接层的宽度（$d_{\text{ff}} = 4d_{\text{model}}$）。我们在训练中使用10%的标签平滑，并且不使用drop-out，进行80次梯度步骤。



(a) 5-layer CNNs (b) ResNet18s (c) Transformers

Figure 13: Scaling of model size with our parameterization of width & embedding dimension.

B.2 Image Classification: Experimental Setup

We describe the details of training for CNNs and ResNets below.

Loss function: Unless stated otherwise, we use the cross-entropy loss for all the experiments.

Data-augmentation: In experiments where data-augmentation was used, we apply `RandomCrop(32, padding=4)` and `RandomHorizontalFlip`. In experiments with added label noise, the label for all augmentations of a given training sample are given the same label.

Regularization: No explicit regularization like weight decay or dropout was applied unless explicitly stated.

Initialization: We use the default initialization provided by PyTorch for all the layers.

Optimization:

- **Adam:** Unless specified otherwise, learning rate was set at constant to $1e{-}4$ and all other parameters were set to their default PyTorch values.
- **SGD:** Unless specified otherwise, learning rate schedule inverse-square root (defined below) was used with initial learning rate $\gamma_0 = 0.1$ and updates every $L = 512$ gradient steps. No momentum was used.

We found our results are robust to various other natural choices of optimizers and learning rate schedule. We used the above settings because (1) they optimize well, and (2) they do not require experiment-specific hyperparameter tuning, and allow us to use the same optimization across many experiments.

Batch size: All experiments use a batchsize of 128.

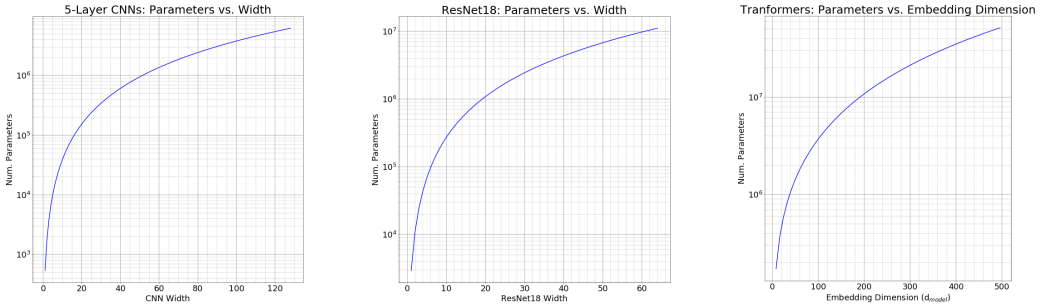
Learning rate schedule descriptions:

- **Inverse-square root** (γ_0, L): At gradient step t, the learning rate is set to $\gamma(t) := \frac{\gamma_0}{\sqrt{1+\lfloor t/512 \rfloor}}$. We set learning-rate with respect to number of gradient steps, and not epochs, in order to allow comparison between experiments with varying train-set sizes.
- **Dynamic drop** (γ_0**, drop, patience**): Starts with an initial learning rate of γ_0 and drops by a factor of 'drop' if the training loss has remained constant or become worse for 'patience' number of gradient steps.

B.3 Neural Machine Translation: Experimental Setup

Here we describe the experimental setup for the neural machine translation experiments.

Training procedure.



(a) 5-layer CNNs (b) ResNet18s (c) Transformers

图 13：缩放 模型大小与我们的宽度和嵌入参数化 维度。

B.2 图像分类：实验设置

我们在下面描述了CNNs和ResNets的训练细节。

损失函数：除非另有说明，否则我们在所有实验中使用交叉熵损失。

数据增强：在使用数据增强的实验中，我们应用了RandomCrop(32, padding=4)和RandomHorizontalFlip。在添加标签噪声的实验中，给定训练样本的所有增强数据都被赋予相同的标签。

正则化：除非明确说明，否则不应用诸如权重衰减或dropout之类的显式正则化。

初始化：我们对所有层使用PyTorch提供的默认初始化。

优化：

- Adam：除非另有说明，学习率被设置为常数1e−4，所有其他参数均设置为其默认的PyTorch值。
- SGD：除非另有说明，使用逆平方根学习率调度（定义如下），初始学习率为$\gamma_0 = 0.1$，每$L = 512$个梯度步骤更新一次。不使用动量。

我们发现我们的结果对各种其他自然选择的优化器和学习率计划具有鲁棒性。我们使用上述设置是因为 (1) 它们优化得很好，并且 (2) 它们不需要特定实验的超参数调整，使我们能够在许多实验中使用相同的优化。

批量大小：所有实验使用的批量大小为128。

学习率计划描述：

- 逆平方根 (γ_0, L)：在梯度步骤 t 时，学习率设置为 $\gamma(t) := \frac{\gamma_0}{\sqrt{1+\lfloor t/512 \rfloor}}$。我们根据梯度步骤的数量而不是周期数来设置学习率，以便在训练集大小不同的实验之间进行比较。
- 动态下降 (γ_0, drop, patience)：以初始学习率 γ_0 开始，如果训练损失在 'patience' 个梯度步骤中保持不变或变得更糟，则按 'drop' 因素下降。

B.3 神经机器翻译：实验设置

在此我们描述神经机器翻译实验的实验设置。

训练过程。

In this setting, the distribution $\mathcal{D}$ consists of triples

$$(x, y, i) \ : \ x \in V_{src}^*, \ y \in V_{tgt}^*, \ i \in \{0, \ldots, |y|\}$$

where V_{src} and V_{tgt} are the source and target vocabularies, the string x is a sentence in the source language, y is its translation in the target language, and i is the index of the token to be predicted by the model. We assume that $i|x, y$ is distributed uniformly on $\{0, \ldots, |y|\}$.

A standard probabilistic model defines an autoregressive factorization of the likelihood:

$$p_M(y|x) = \prod_{i=1}^{|y|} p_M(y_i|y_{<i}, x).$$

Given a set of training samples S, we define

$$\text{Error}_S(M) = \frac{1}{|S|} \sum_{(x,y,i) \in S} -\log p_M(y_i|y_{<i}, x).$$

In practice, S is *not* constructed from independent samples from D, but rather by first sampling (x, y) and then including all $(x, y, 0), \ldots, (x, y, |y|)$ in S.

For training transformers, we replicate the optimization procedure specified in Vaswani et al. (2017) section 5.3, where the learning rate schedule consists of a "warmup" phase with linearly increasing learning rate followed by a phase with inverse square-root decay. We preprocess the data using byte pair encoding (BPE) as described in Sennrich et al. (2015). We use the implementation provided by fairseq (`https://github.com/pytorch/fairseq`).

Datasets. The IWSLT '14 German to English dataset contains TED Talks as described in Cettolo et al. (2012). The WMT '14 English to French dataset is taken from `http://www.statmt.org/wmt14/translation-task.html`.

B.4 Per-section Experimental Details

Here we provide full details for experiments in the body, when not otherwise provided.

Introduction: Experimental Details Figure 1: All models were trained using Adam with learning-rate 0.0001 for 4K epochs. Plotting means and standard deviations for 5 trials, with random network initialization.

Model-wise Double Descent: Experimental Details Figure 7: Plotting means and standard deviations for 5 trials, with random network initialization.

Sample-wise Nonmonotonicity: Experimental Details Figure 11a: All models are trained with SGD for 500K epochs, and data-augmentation. Bottom: Means and standard deviations from 5 trials with random initialization, and random subsampling of the train set.

在此设置中，分布 $\mathcal{D}$ 由三元组组成

$$(x, y, i) \ : \ x \in V_{src}^*, \ y \in V_{tgt}^*, \ i \in \{0, \ldots, |y|\}$$

其中 V_{src} 和 V_{tgt} 是源语言和目标语言的词汇表，字符串 x 是源语言中的一个句子，y 是其在目标语言中的翻译，i 是模型要预测的标记的索引。我们假设 $i|x, y$ 在 $\{0, \ldots, |y|\}$ 上均匀分布。

标准概率　　概率模型定义了似然的自回归分解　　可能性：

$$p_M(y|x) = \prod_{i=1}^{|y|} p_M(y_i|y_{<i}, x).$$

给定一组训练样本 S，我们定义

$$\text{Error}_S(M) = \frac{1}{|S|} \sum_{(x,y,i)\in S} -\log p_M(y_i|y_{<i}, x).$$

在实践中，S 是从 D 的独立样本构建的 *not*，而是通过首先采样 (x, y)，然后将所有 $(x, y, 0)$ $, \ldots, (x, y, |y|)$ 包含在 S 中。

为了训练transformers，我们复制了Vaswani等人（2017）第5.3节中指定的优化过程，其中学习率调度包括一个“ 预热 ”阶段，学习率线性增加，随后是一个逆平方根衰减阶段。我们使用Sennrich等人（2015）中描述的字节对编码（BPE）对数据进行预处理。我们使用了fairseq（https://github.com/pytorch/fairseq）提供的实现。

数据集。IWSLT '14 德语到英语数据集包含 TED 演讲，如 Cettolo 等人 (2012) 所述。WMT '14 英语到法语数据集取自 http://www.statmt.org/wmt14/translation-task.html。

B.4 每节实验细节

在此，我们提供正文中实验的详细信息，除非另有说明。

介绍：实验细节 图1：所有模型均使用Adam优化器进行训练，学习率为0.0001，训练4K个周期。绘制5次试验的均值和标准差，随机网络初始化。

模型双重下降：实验细节 图7：绘制5次试验的均值和标准差，随机网络初始化。

样本非单调性：实验细节 图11a：所有模型均使用SGD训练500K个周期，并进行数据增强。底部：5次随机初始化和随机子采样训练集的试验的均值和标准差。

C EXTENDED DISCUSSION OF RELATED WORK

Belkin et al. (2018): This paper proposed, in very general terms, that the apparent contradiction between traditional notions of the bias-variance trade-off and empirically successful practices in deep learning can be reconciled under a double-descent curve—as model complexity increases, the test error follows the traditional "U-shaped curve", but beyond the point of interpolation, the error starts to *decrease*. This work provides empirical evidence for the double-descent curve with fully connected networks trained on subsets of MNIST, CIFAR10, SVHN and TIMIT datasets. They use the l_2 loss for their experiments. They demonstrate that neural networks are not an aberration in this regard—double-descent is a general phenomenon observed also in linear regression with random features and random forests.

Theoretical works on linear least squares regression: A variety of papers have attempted to theoretically analyze this behavior in restricted settings, particularly the case of least squares regression under various assumptions on the training data, feature spaces and regularization method.

1. Advani & Saxe (2017); Hastie et al. (2019) both consider the linear regression problem stated above and analyze the generalization behavior in the asymptotic limit $N, D \to \infty$ using random matrix theory. Hastie et al. (2019) highlight that when the model is misspecified, the minimum of training error can occur for over-parameterized models
2. Belkin et al. (2019) Linear least squares regression for two data models, where the input data is sampled from a Gaussian and a Fourier series model for functions on a circle. They provide a finite-sample analysis for these two cases
3. Bartlett et al. (2019) provides generalization bounds for the minimum l_2-norm interpolant for Gaussian features
4. Muthukumar et al. (2019) characterize the fundamental limit of of any interpolating solution in the presence of noise and provide some interesting Fourier-theoretic interpretations.
5. Mei & Montanari (2019): This work provides asymptotic analysis for ridge regression over random features

Similar double descent behavior was investigated in Opper (1995; 2001)

Geiger et al. (2019b) showed that deep fully connected networks trained on the MNIST dataset with hinge loss exhibit a "jamming transition" when the number of parameters exceeds a threshold that allows training to near-zero train loss. Geiger et al. (2019a) provide further experiments on CIFAR-10 with a convolutional network. They also highlight interesting behavior with ensembling around the critical regime, which is consistent with our informal intuitions in Section 5 and our experiments in Figures 28, 29.

Advani & Saxe (2017); Geiger et al. (2019b;a) also point out that double-descent is not observed when optimal early-stopping is used.

C EXTENDED DISCUSSION OF RELATED WORK

Belkin 等人 (2018)：本文在非常一般的意义上提出，传统的偏差-方差权衡概念与深度学习中经验上成功的实践之间的明显矛盾可以在双下降曲线下得到调和——随着模型复杂性的增加，测试误差遵循传统的“ U形曲线 ”，但在插值点之后，误差开始*decrease*。这项工作提供了在MNIST、CIFAR10、SVHN和TIMIT数据集的子集上训练的全连接网络的双下降曲线的实证证据。他们在实验中使用了l_2损失。他们证明神经网络在这方面并不是一个例外——双下降是一个普遍现象，也在具有随机特征的线性回归和随机森林中观察到。

线性最小二乘回归的理论研究：各种论文试图在受限环境中对这种行为进行理论分析，特别是在对训练数据、特征空间和正则化方法进行各种假设的情况下的最小二乘回归。

> 1. Advani & Saxe (2017); Hastie 等人 (2019) 都考虑了上述线性回归问题，并使用随机矩阵理论分析了在渐近极限 $N, D \to \infty$ 下的泛化行为。Hastie 等人 (2019) 强调，当模型被错误指定时，训练误差的最小值可能出现在过度参数化的模型中。2. Belkin 等人 (2019) 针对两种数据模型的线性最小二乘回归，其中输入数据从高斯和傅里叶级数模型中采样，用于圆上的函数。他们为这两种情况提供了有限样本分析。3. Bartlett 等人 (2019) 为高斯特征的最小 l_2-范数插值提供了泛化界限。4. Muthukumar 等人 (2019) 描述了在存在噪声的情况下任何插值解的基本极限，并提供了一些有趣的傅里叶理论解释。5. Mei & Montanari (2019): 这项工作为随机特征上的岭回归提供了渐近分析。

类似的双重下降行为在 Opper (1995; 2001) 中进行了研究。

Geiger 等人 (2019b) 表明，在 MNIST 数据集上使用铰链损失训练的深度全连接网络在参数数量超过允许训练到接近零训练损失的阈值时会出现“ 堵塞转变 ”。Geiger 等人 (2019a) 在 CIFAR-10 上使用卷积网络进行了进一步的实验。他们还强调了在关键状态下集成的有趣行为，这与我们在第 5 节中的非正式直觉以及我们在图 28 和图 29 中的实验一致。

Advani & Saxe (2017); Geiger 等人 (2019b;a) 也指出，当使用最佳提前停止时，不会观察到双重下降。

D RANDOM FEATURES: A CASE STUDY

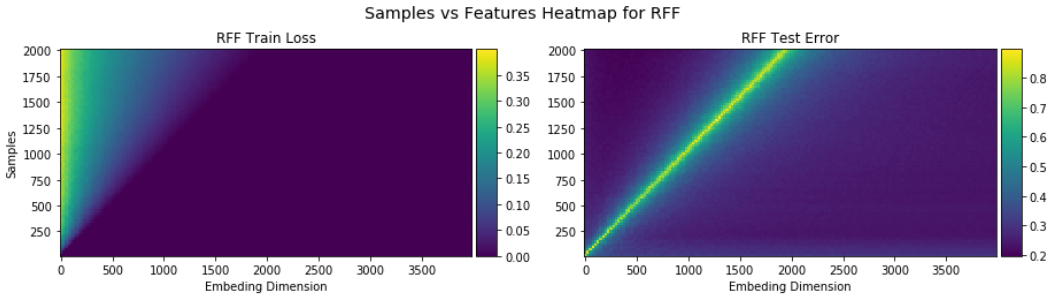


Figure 14: **Random Fourier Features** on the Fashion MNIST dataset. The setting is equivalent to two-layer neural network with e^{-ix} activation, with randomly-initialized first layer that is fixed throughout training. The second layer is trained using gradient flow.

In this section, for completeness sake, we show that both the model- and sample-wise double descent phenomena are not unique to deep neural networks—they exist even in the setting of Random Fourier Features of Rahimi & Recht (2008). This setting is equivalent to a two-layer neural network with e^{-ix} activation. The first layer is initialized with a $\mathcal{N}(0, \frac{1}{d})$ Gaussian distribution and then fixed throughout training. The width (or embedding dimension) d of the first layer parameterizes the model size. The second layer is initialized with 0s and trained with MSE loss.

Figure 14 shows the grid of Test Error as a function of both number of samples n and model size d. Note that in this setting $\mathrm{EMC} = d$ (the embedding dimension). As a result, as demonstrated in the figure, the peak follows the path of $n = d$. Both model-wise and sample-wise (see figure 15) double descent phenomena are captured, by horizontally and vertically crossing the grid, respectively.

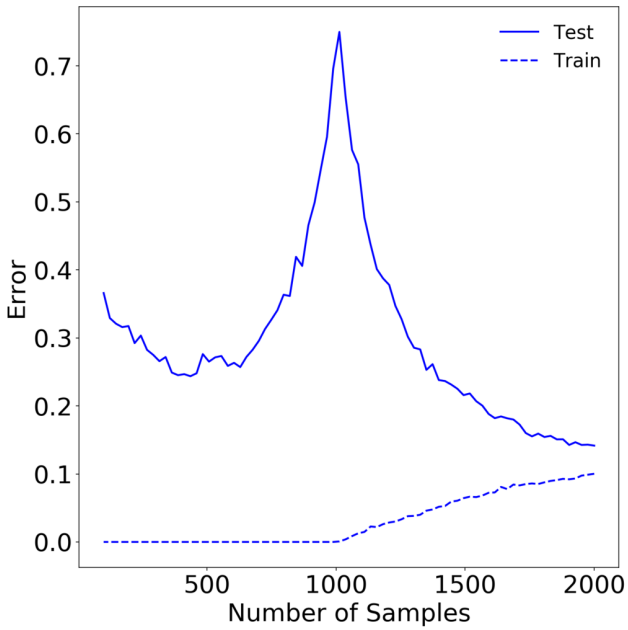


Figure 15: Sample-wise double-descent slice for Random Fourier Features on the Fashion MNIST dataset. In this figure the embedding dimension (number of random features) is 1000.

D RANDOM FEATURES: A CASE STUDY

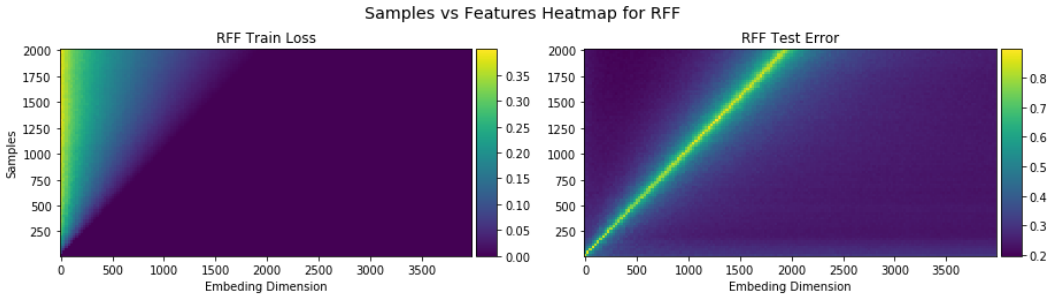


图 14：Fashion MNIST 数据集上的随机傅里叶特征。该设置等同于具有 e^{-ix} 激活的两层神经网络，第一层随机初始化并在整个训练过程中保持不变。第二层使用梯度流进行训练。

在本节中，为了完整性，我们展示了模型和样本的双重下降现象并非深度神经网络所独有——它们甚至存在于Rahimi & Recht (2008)的随机傅里叶特征设置中。该设置等同于一个具有e^{-ix}激活的两层神经网络。第一层用$\mathcal{N}(0, \frac{1}{d})$高斯分布初始化，然后在整个训练过程中保持不变。第一层的宽度（或嵌入维度）d参数化模型大小。第二层用0初始化，并使用MSE损失进行训练。

图14显示了测试误差的网格，作为样本数量n和模型大小d的函数。请注意，在此设置中，EMC $= d$ (嵌入维度)。因此，如图所示，峰值沿着$n = d$的路径。通过水平和垂直穿过网格，分别捕捉到了模型层面和样本层面的双重下降现象（见图15）。

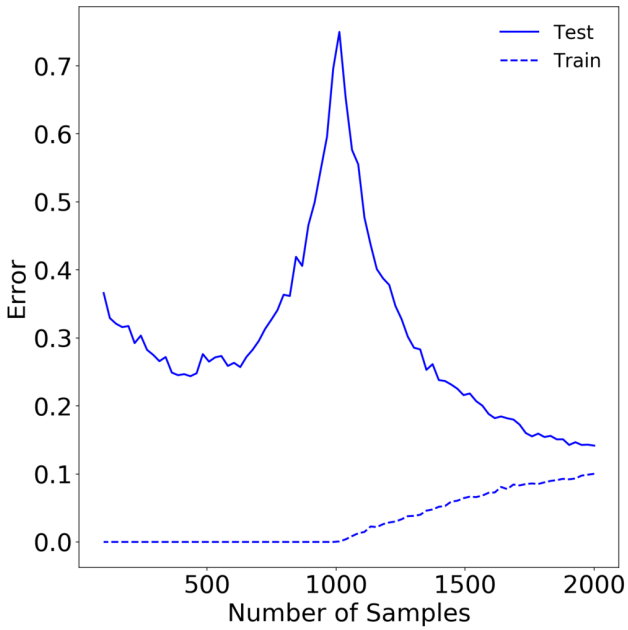
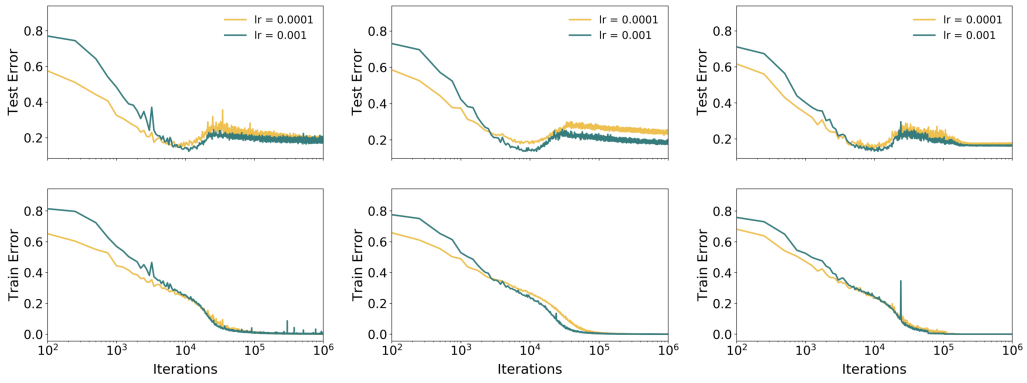


图15：Fashion MNIST数据集上随机傅里叶特征的样本双重下降切片。在此图中，嵌入维度（随机特征的数量）为1000。

E Appendix: Additional Experiments

E.1 Epoch-wise Double Descent: Additional results

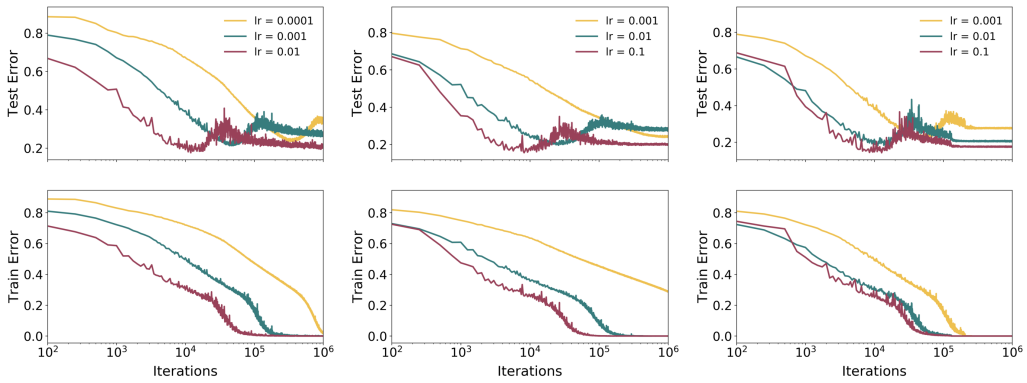
Here, we provide a rigorous evaluation of epoch-wise double descent for a variety of optimizers and learning rate schedules. We train ResNet18 on CIFAR-10 with data-augmentation and 20% label noise with three different optimizers—Adam, SGD, SGD + Momentum (momentum set to 0.9) and three different learning rate schedules—constant, inverse-square root, dynamic drop for differnet values of initial learning rate. We observe that double-descent occurs reliably for all optimizers and learning rate schedules and the peak of the double descent curve shifts with the interpolation point.



(a) Constant learning rate (b) Inverse-square root learning rate (c) Dynamic learning rate

Figure 16: **Epoch-wise double descent** for ResNet18 trained with Adam and multiple learning rate schedules

A practical recommendation resulting from epoch-wise double descent is that stopping the training when the test error starts to increase may not always be the best strategy. In some cases, the test error may decrease again after reaching a maximum, and the final value may be lower than the minimum earlier in training.



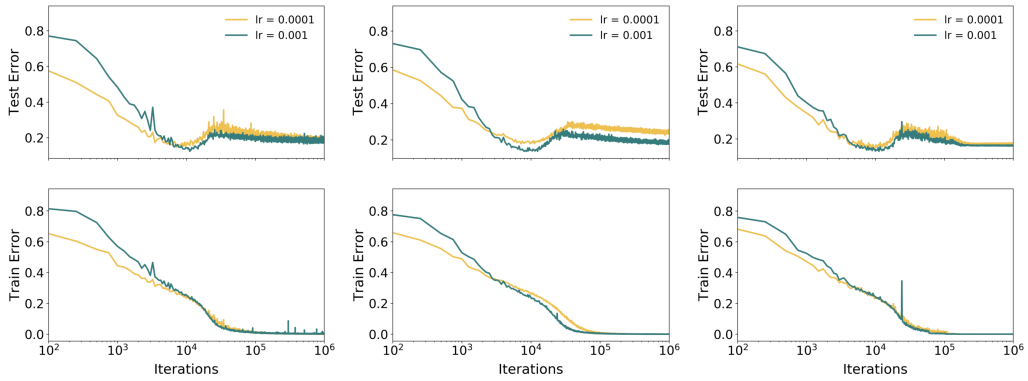
(a) Constant learning rate (b) Inverse-square root learning rate (c) Dynamic learning rate

Figure 17: **Epoch-wise double descent** for ResNet18 trained with SGD and multiple learning rate schedules

E APPENDIX: ADDITIONAL EXPERIMENTS

E.1 逐纪元双重下降：附加结果

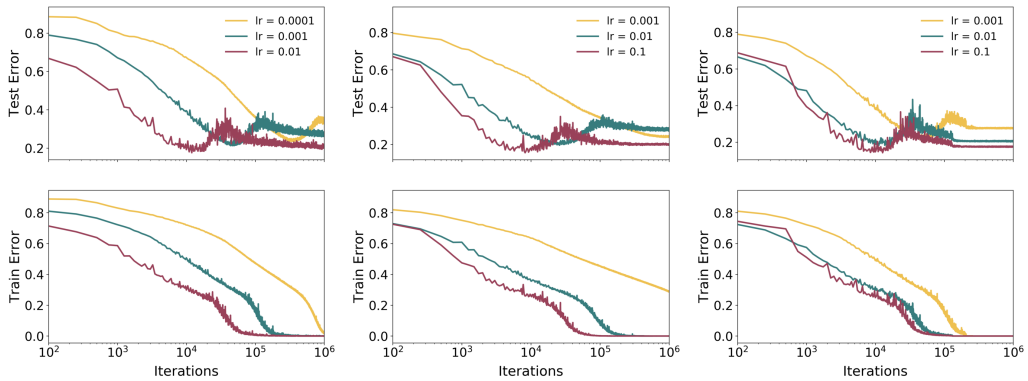
在这里，我们对多种优化器和学习率计划进行了逐时代双重下降的严格评估。我们在 CIFAR-10 上训练 ResNet18，使用数据增强和 20% 的标签噪声，并使用三种不同的优化器——Adam、SGD、SGD + 动量（动量设置为 0.9）以及三种不同的学习率计划——恒定、反平方根、动态下降，针对不同的初始学习率值。我们观察到，对于所有优化器和学习率计划，双重下降现象都可靠地发生，并且双重下降曲线的峰值随着插值点而移动。



(a) Constant learning rate (b) Inverse-square root learning rate (c) Dynamic learning rate

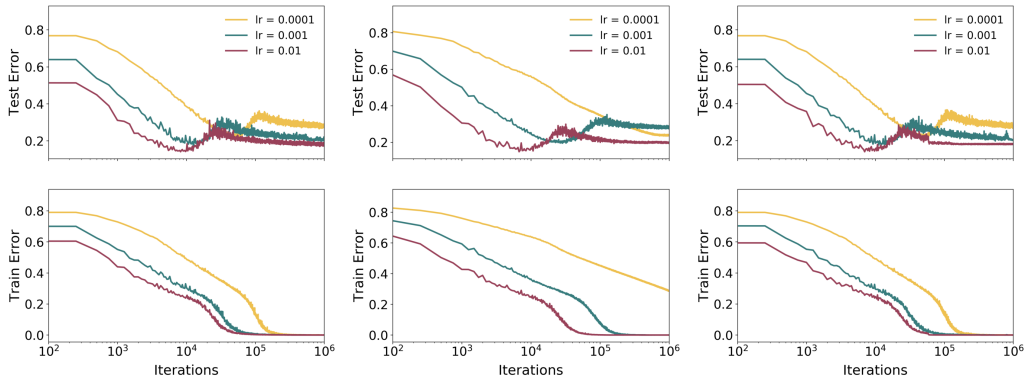
图16：使用Adam和多种学习率计划训练的ResNet18的逐周期双重下降

来自历元双重下降的一个实际建议是，当测试误差开始增加时停止训练可能并不总是最佳策略。在某些情况下，测试误差在达到最大值后可能会再次减少，并且最终值可能低于训练早期的最小值。



(a) Constant learning rate (b) Inverse-square root learning rate (c) Dynamic learning rate

图 17：使用 SGD 和多种学习率计划训练的 ResNet18 的逐周期双重下降



(a) Constant learning rate (b) Inverse-square root learning rate (c) Dynamic learning rate

Figure 18: **Epoch-wise double descent** for ResNet18 trained with SGD+Momentum and multiple learning rate schedules

E.2 Model-Wise Double Descent: Additional Results

E.2.1 Clean Settings With Model-wise Double Descent

CIFAR100, ResNet18

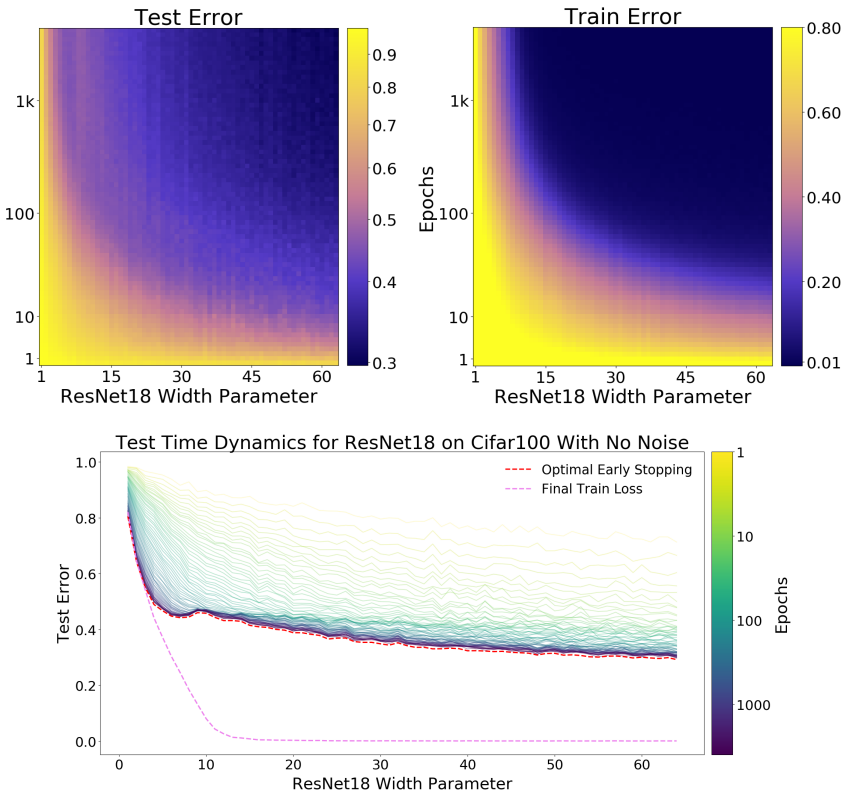
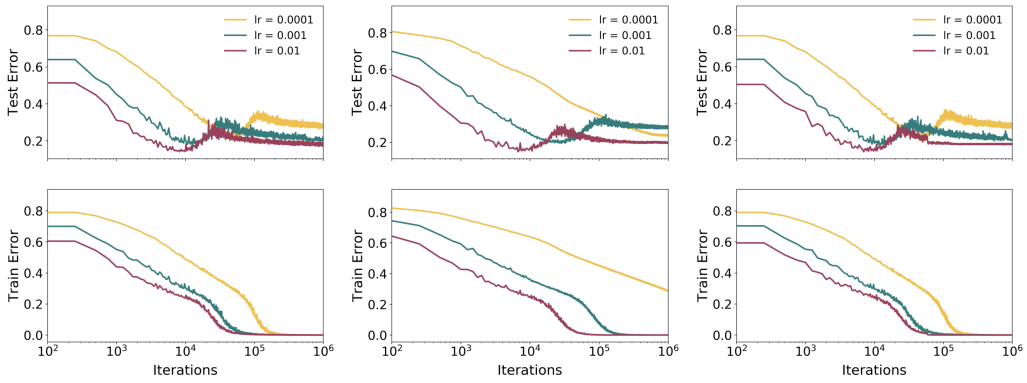


Figure 19: **Top:** Train and test performance as a function of both model size and train epochs. **Bottom:** Test error dynamics of the same model (ResNet18, on CIFAR-100 with no label noise, data-augmentation and Adam optimizer trained for 4k epochs with learning rate 0.0001). Note that even with optimal early stopping this setting exhibits double descent.



(a) Constant learning rate (b) Inverse-square root learning rate (c) Dynamic learning rate

图 18：使用 SGD+动量和多种学习率计划训练的 ResNet18 的逐周期双重下降

E.2 模型层面的双重下降：附加结果

E.2.1 具有模型双重下降的清洁设置

CIFAR100，ResNet18

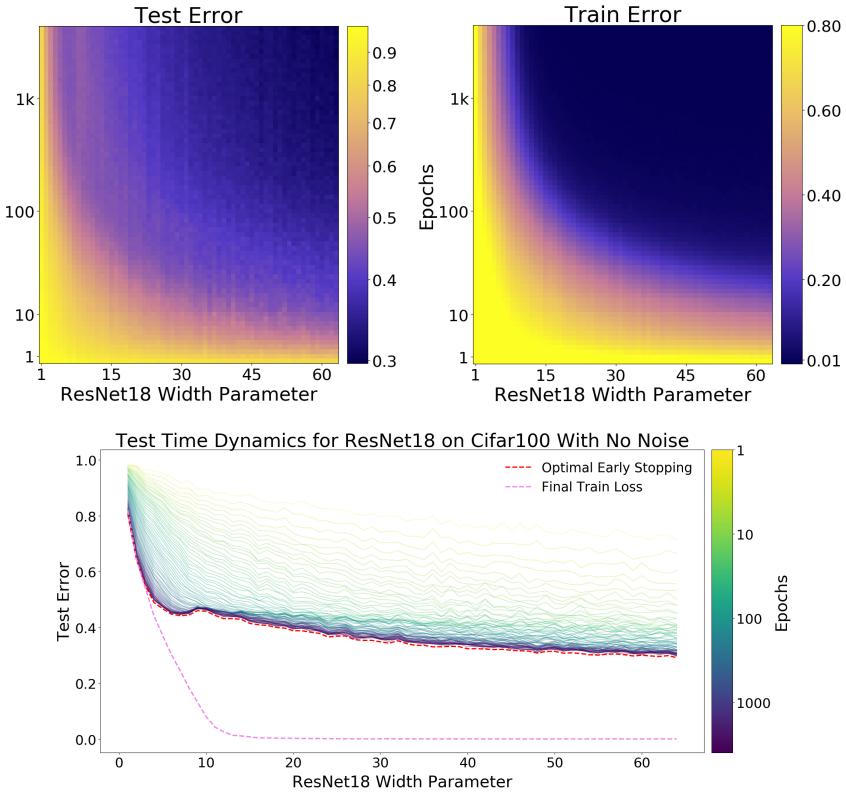


图19：顶部：训练和测试性能作为模型大小和训练周期的函数。底部：相同模型的测试误差动态（ResNet18，在CIFAR-100上无标签噪声，数据增强和Adam优化器，训练4k周期，学习率为0.0001）。注意，即使在最佳提前停止的情况下，此设置也表现出双重下降。

CIFAR100, Standard CNN

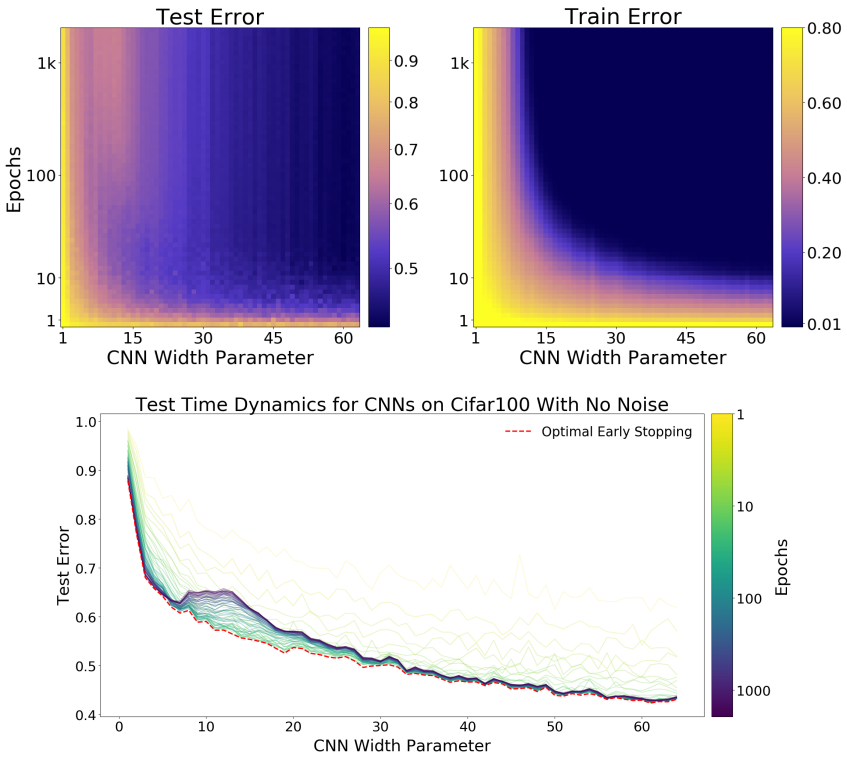


Figure 20: **Top:** Train and test performance as a function of both model size and train epochs. **Bottom:** Test error dynamics of the same models. 5-Layer CNNs, CIFAR-100 with no label noise, no data-augmentation Trained with SGD for 1e6 steps. Same experiment as Figure 7.

CIFAR100, Standard CNN

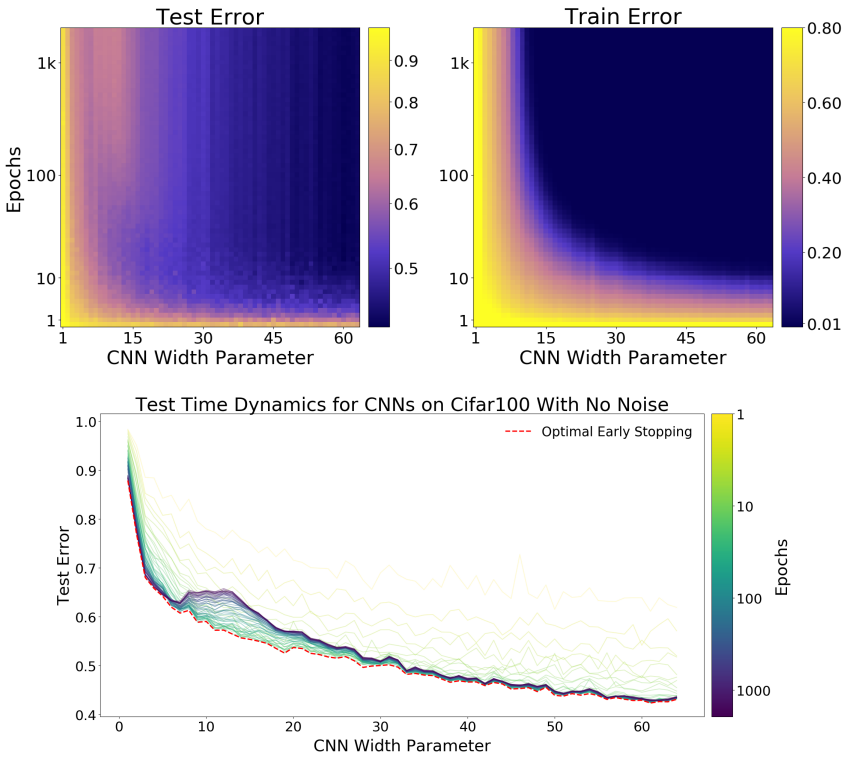


图 20：顶部：训练和测试性能作为模型大小和训练周期的函数。底部：相同模型的测试误差动态。5 层 CNN，CIFAR-100 无标签噪声，无数据增强，使用 SGD 训练 1e6 步。与图 7 相同的实验。

E.2.2 WEIGHT DECAY

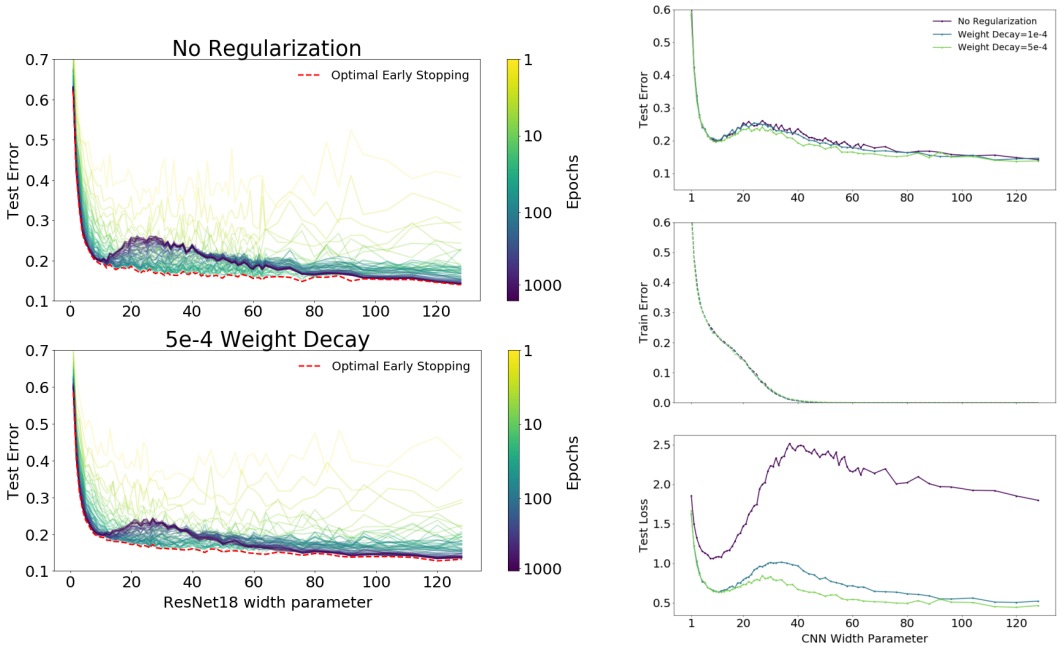
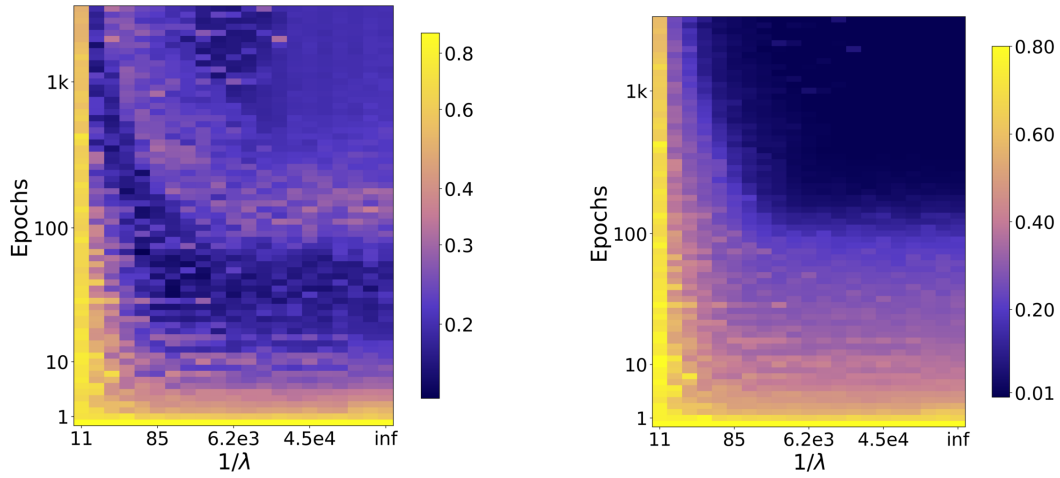


Figure 21: **Left:** Test error dynamics with weight decay of 5e-4 (bottom left) and without weight decay (top left). **Right:** Test and train error and *test loss* for models with varying amounts of weight decay. All models are 5-Layer CNNs on CIFAR-10 with 10% label noise, trained with data-augmentation and SGD for 500K steps.

Here, we now study the effect of varying the level of regularization on test error. We train CIFAR10 with data-augmentation and 20% label noise on ResNet18 for weight decay co-efficients λ ranging from 0 to 0.1. We train the networks using SGD + inverse-square root learning rate. Figure below shows a picture qualitatively very similar to that observed for model-wise double descent wherein "model complexity" is now controlled by the regularization parameter. This confirms our generalized double descent hypothesis along yet another axis of Effective Model Complexity.



(a) Test Error

(b) Train Loss

Figure 22: Generalized double descent for weight decay. We found that using the same initial learning rate for all weight decay values led to training instabilities. This resulted in some noise in the Test Error (Weight Decay $\times$ Epochs) plot shown above.

E.2.2 WEIGHT DECAY

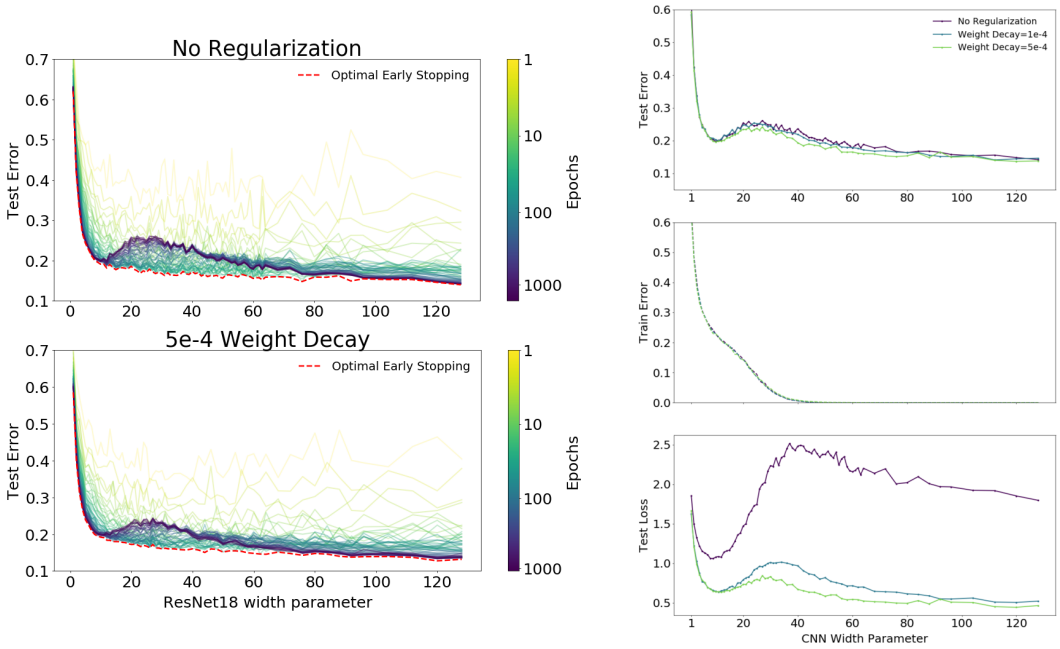
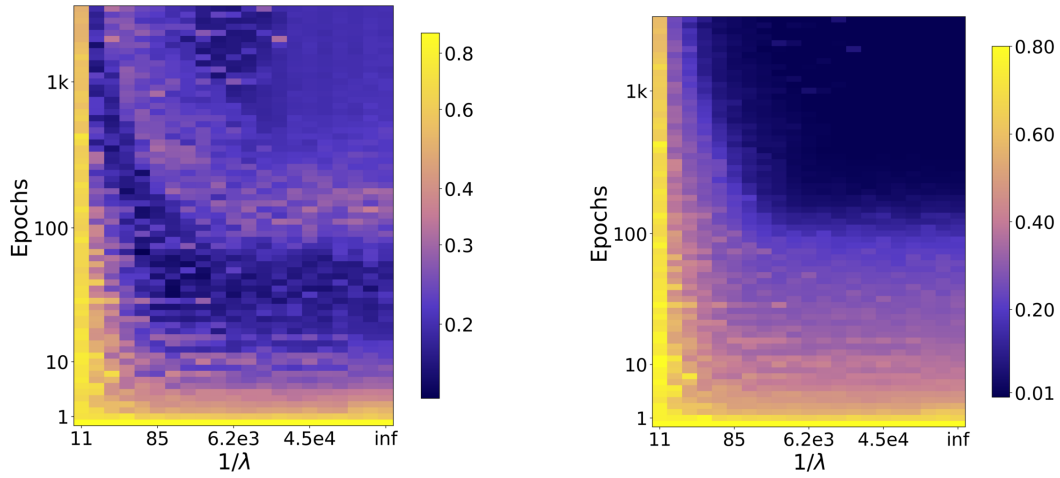


图21：左图：权重衰减为5e-4（左下）和无权重衰减（左上）的测试误差动态。右图：具有不同权重衰减量的模型的测试和训练误差以及*test loss*。所有模型都是在CIFAR-10上带有10%标签噪声的5层CNN，使用数据增强和SGD训练500K步。

在这里，我们研究正则化水平变化对测试误差的影响。我们在ResNet18上训练CIFAR10，使用数据增强和20%的标签噪声，权重衰减系数λ范围从0到0.1。我们使用SGD +反平方根学习率训练网络。下图显示的图像在质量上与模型双重下降观察到的非常相似，其中“模型复杂性”现在由正则化参数控制。这证实了我们的广义双重下降假设在有效模型复杂性的另一个轴上的正确性。



(a) Test Error (b) Train Loss

图 22：权重衰减的广义双重下降。我们发现，对所有权重衰减值使用相同的初始学习率会导致训练不稳定。这导致了上图中测试误差（权重衰减 × 轮次）图中的一些噪声。

E.2.3 EARLY STOPPING DOES NOT EXHIBIT DOUBLE DESCENT

Language models

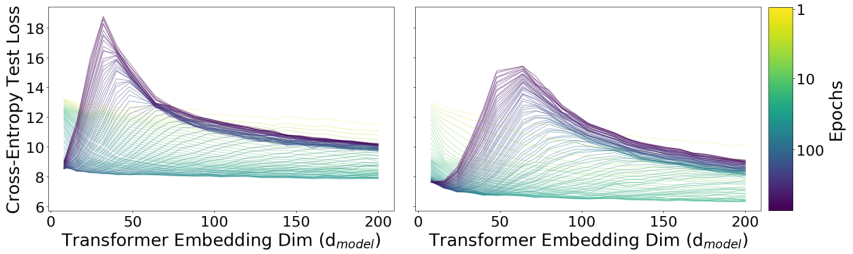


Figure 23: Model-wise test error dynamics for a subsampled IWSLT'14 dataset. Left: 4k samples, Right: 18k samples. Note that with optimal early-stopping, more samples is always better.

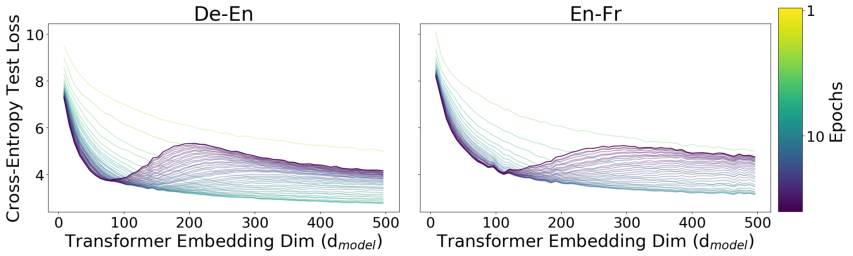


Figure 24: Model-wise test error dynamics for a IWSLT'14 de-en and subsampled WMT'14 en-fr datasets. **Left**: IWSLT'14, **Right**: subsampled (200k samples) WMT'14. Note that with optimal early-stopping, the test error is much lower for this task.

CIFAR10, 10% noise, SGD

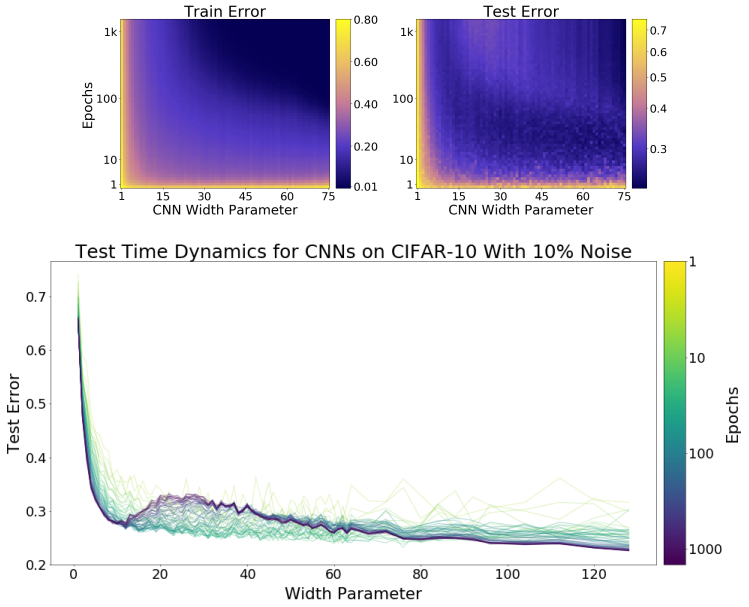


Figure 25: **Top:** Train and test performance as a function of both model size and train epochs. **Bottom:** Test error dynamics of the same model (CNN, on CIFAR-10 with 10% label noise, data-augmentation and SGD optimizer with learning rate $\propto 1/\sqrt{T}$).

E.2.3 EARLY STOPPING DOES NOT EXHIBIT DOUBLE DESCENT

语言模型

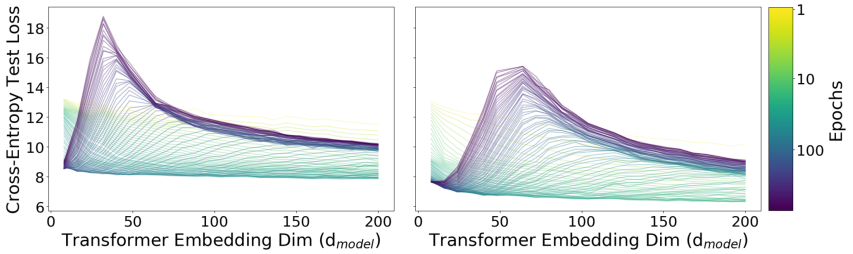


图 23：对 IWSLT ‘ 14 数据集进行子采样的模型测试误差动态。左：4k 样本，右：18k 样本。请注意，使用最佳的提前停止策略时，更多的样本总是更好。

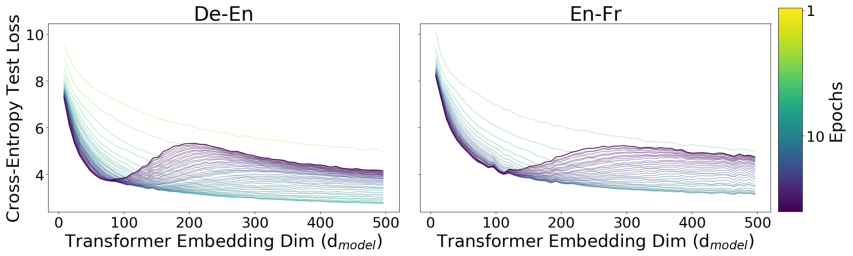


图 24：IWSLT ‘ 14 de-en 和子采样 WMT ‘ 14 en-fr 数据集的模型测试误差动态。左：IWSLT ‘ 14，右：子采样（200k 样本）WMT ‘ 14。请注意，通过最佳的提前停止策略，该任务的测试误差要低得多。

CIFAR10，10%噪声，SGD

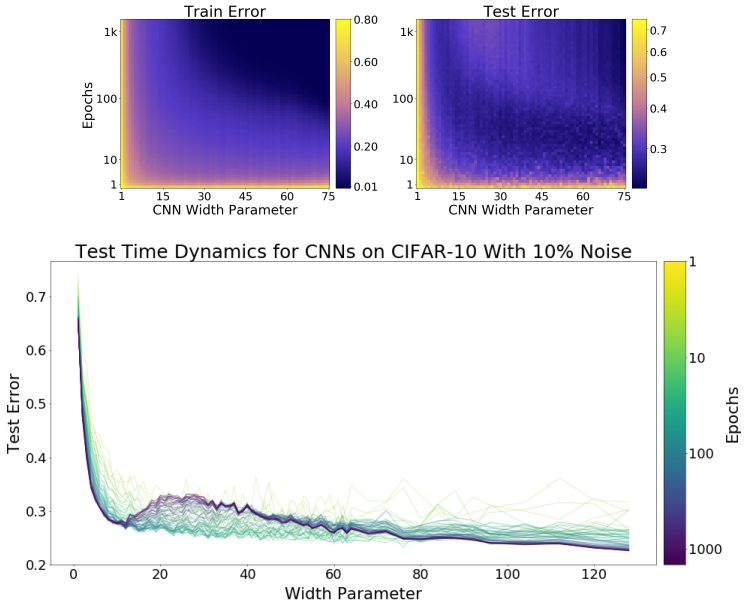


图 25：顶部：训练和测试性能作为模型大小和训练周期的函数。底部：相同模型的测试误差动态（CNN，在 CIFAR-10 上，标签噪声为 10%，数据增强和 SGD 优化器，学习率为 $\propto 1/\sqrt{T}$）。

E.2.4 TRAINING PROCEDURE

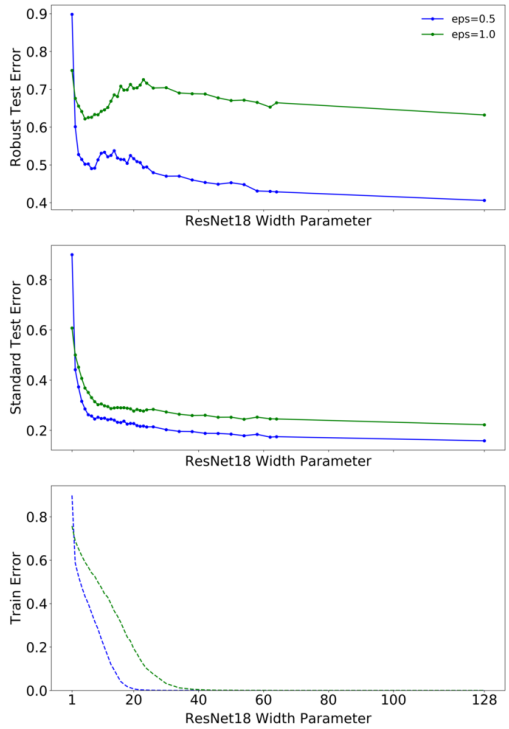


Figure 26: **Model-wise double descent for adversarial training** ResNet18s on CIFAR-10 (subsampled to 25k train samples) with no label noise. We train for L2 robustness of radius $\epsilon = 0.5$ and $\epsilon = 1.0$, using 10-step PGD (Goodfellow et al. (2014); Madry et al. (2017)). Trained using SGD (batch size 128) with learning rate 0.1 for 400 epochs, then 0.01 for 400 epochs.

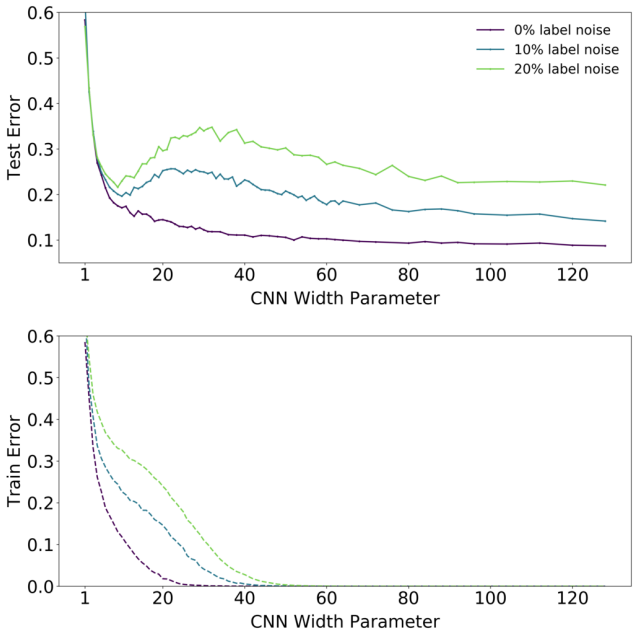


Figure 27

E.2.4 TRAINING PROCEDURE

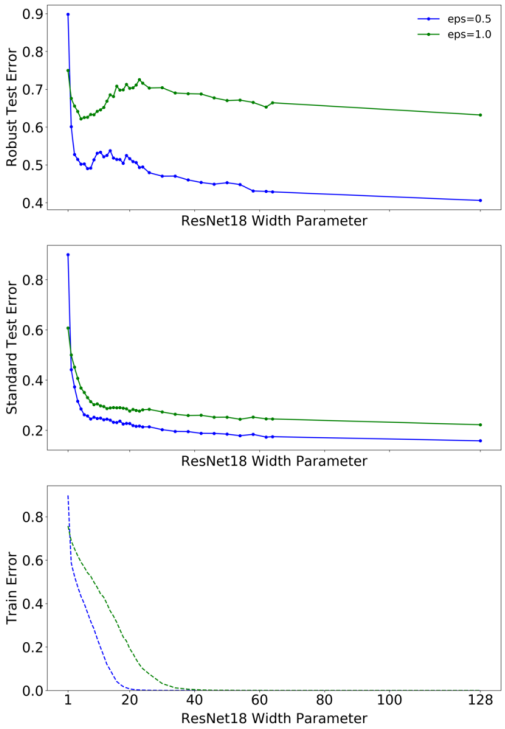


图 26：在 CIFAR-10 数据集（训练样本子采样至 25k）上对 ResNet18 进行对抗训练的模型双重下降，无标签噪声。我们训练 L2 半径为 $\epsilon = 0.5$ 和 $\epsilon = 1.0$ 的鲁棒性，使用 10 步 PGD（Goodfellow 等人，2014；Madry 等人，2017）。使用 SGD（批量大小为 128）进行训练，学习率为 0.1 训练 400 个周期，然后学习率为 0.01 再训练 400 个周期。

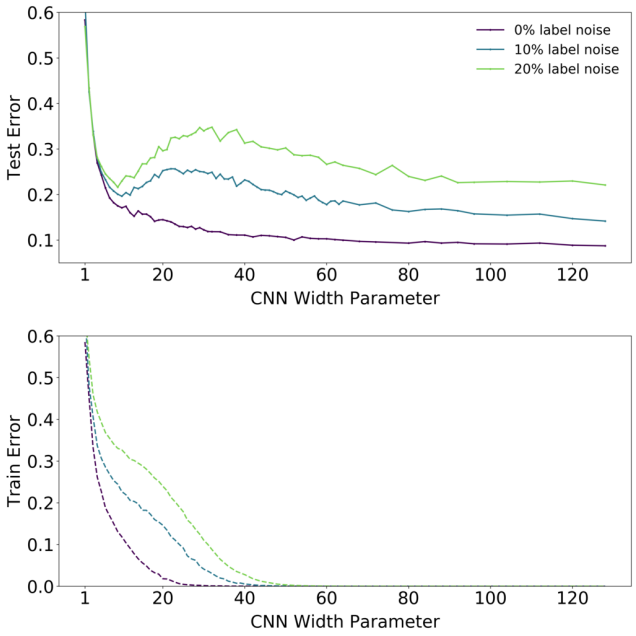


图 27

E.3 ENSEMBLING

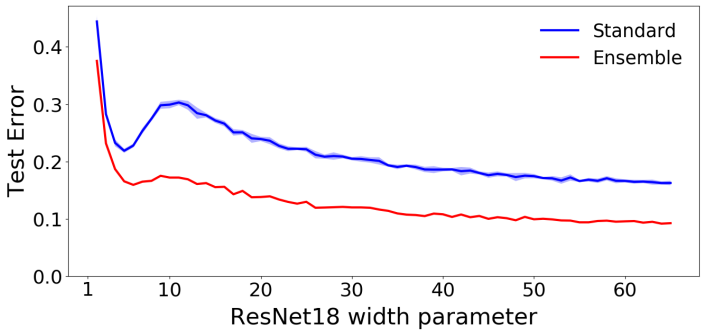


Figure 28: **Effect of Ensembling (ResNets, 15% label noise).** Test error of an ensemble of 5 models, compared to the base models. The ensembled classifier is determined by plurality vote over the 5 base models. Note that emsembling helps most around the critical regime. All models are ResNet18s trained on CIFAR-10 with 15% label noise, using Adam for 4K epochs (same setting as Figure 1). Test error is measured against the original (not noisy) test set, and each model in the ensemble is trained using a train set with independently-sampled 15% label noise.

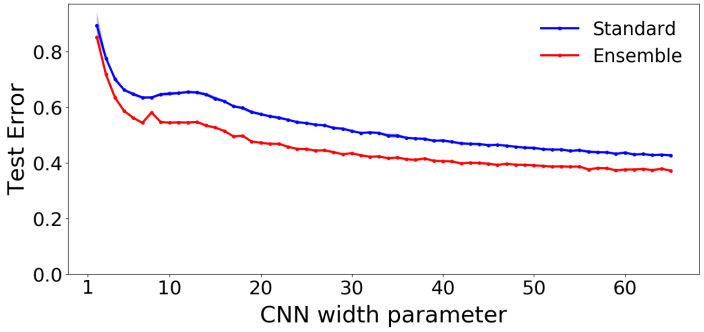


Figure 29: **Effect of Ensembling (CNNs, no label noise).** Test error of an ensemble of 5 models, compared to the base models. All models are 5-layer CNNs trained on CIFAR-10 with no label noise, using SGD and no data augmentation. (same setting as Figure 7).

E.3 ENSEMBLING

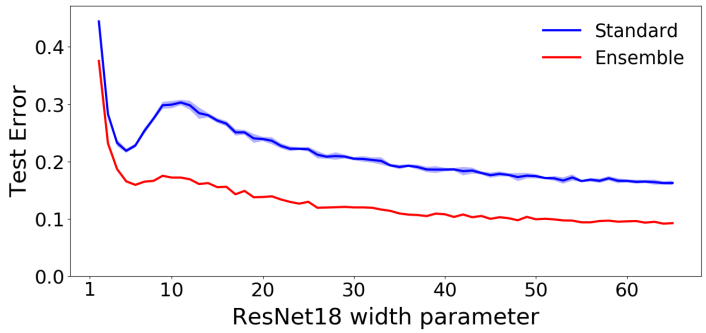


图28：集成效果（ResNets，15%标签噪声）。5个模型的集成测试误差，与基础模型相比。集成分类器通过对5个基础模型的多数投票决定。请注意，集成在关键阶段最为有效。所有模型都是在CIFAR-10上训练的ResNet18，标签噪声为15%，使用Adam优化器训练4K个周期（与图1相同的设置）。测试误差是针对原始（无噪声）测试集测量的，集成中的每个模型都是使用一个独立采样的15%标签噪声的训练集进行训练的。

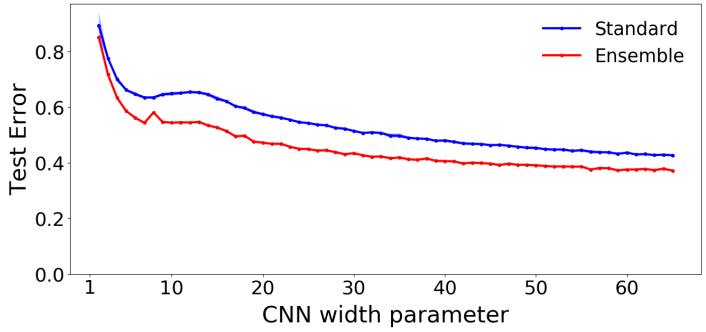


图29：集成效果（CNN，无标签噪声）。5个模型的集成测试误差，与基础模型相比。所有模型都是在CIFAR-10上训练的5层CNN，无标签噪声，使用SGD且无数据增强。（与图7相同的设置）。