Double Descent

Jikai Jin
Thanawat Sornwanee

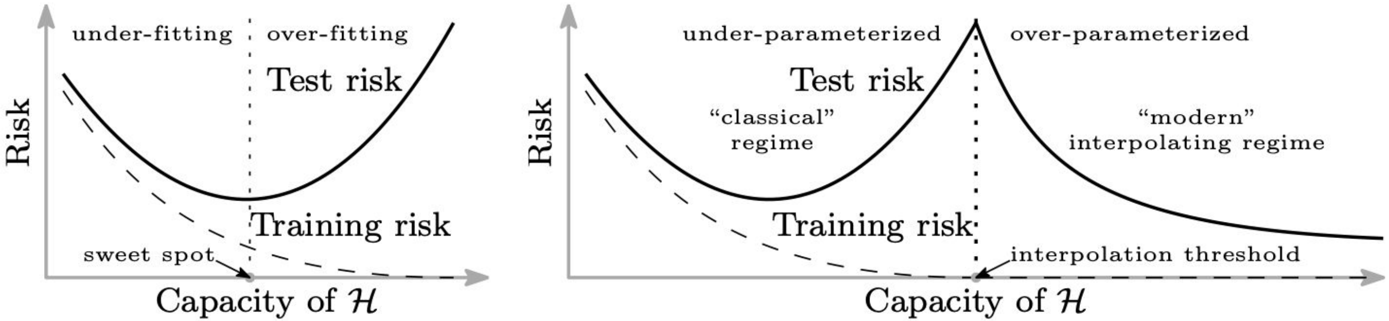
ReFoRM 2026

Double Descent

Reconciling modern machine learning practice and the bias-variance trade-off

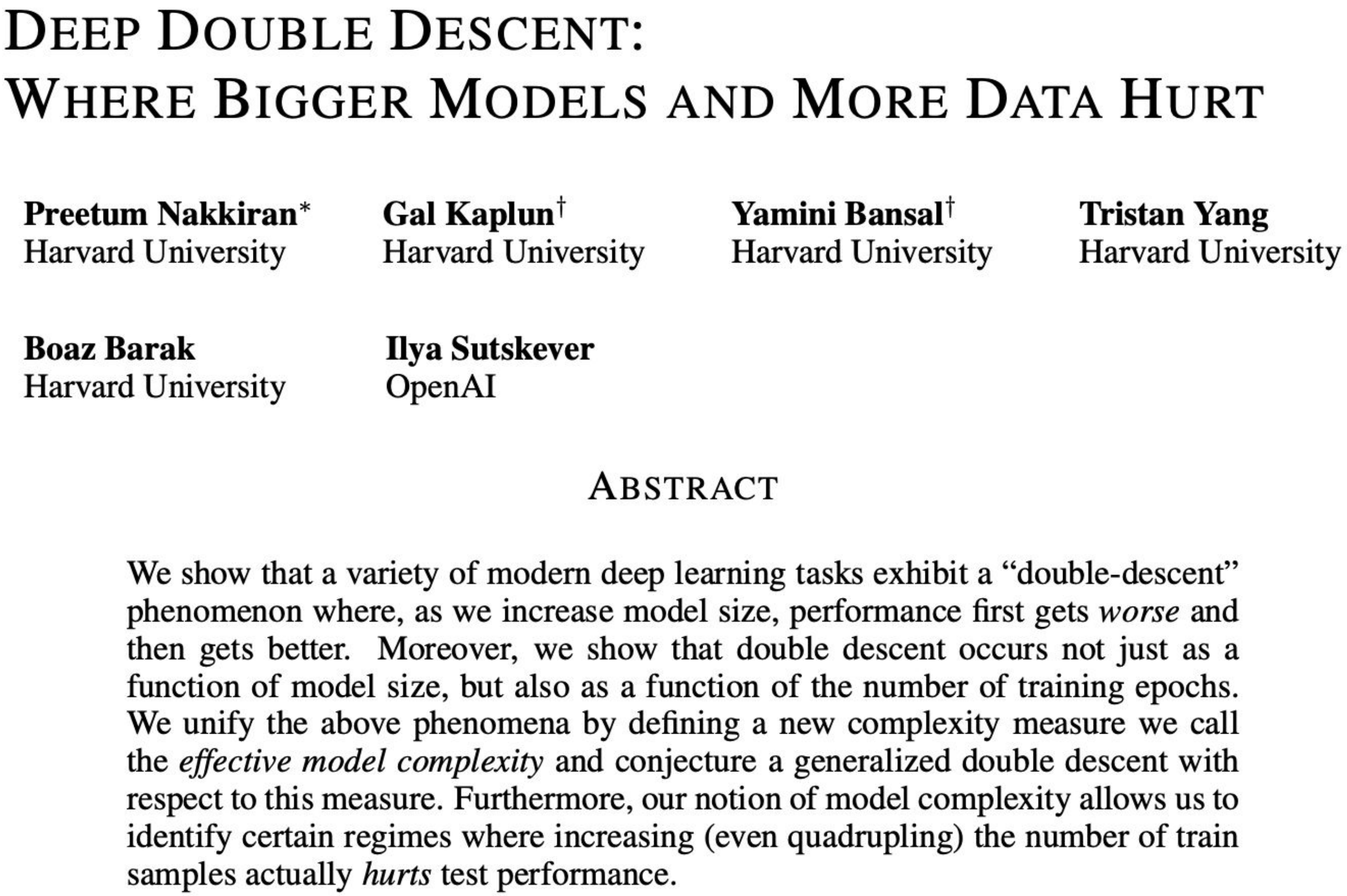
Mikhail Belkin[a], Daniel Hsu[b], Siyuan Ma[a], and Soumik Mandal[a]

[a] *The Ohio State University, Columbus, OH*
[b] *Columbia University, New York, NY*



(a) **(b)**

Figure 1: **Curves for training risk (dashed line) and test risk (solid line).** (**a**) The classical *U-shaped risk curve* arising from the bias-variance trade-off. (**b**) The *double descent risk curve*, which incorporates the U-shaped risk curve (i.e., the "classical" regime) together with the observed behavior from using high capacity function classes (i.e., the "modern" interpolating regime), separated by the interpolation threshold. The predictors to the right of the interpolation threshold have zero training risk.



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

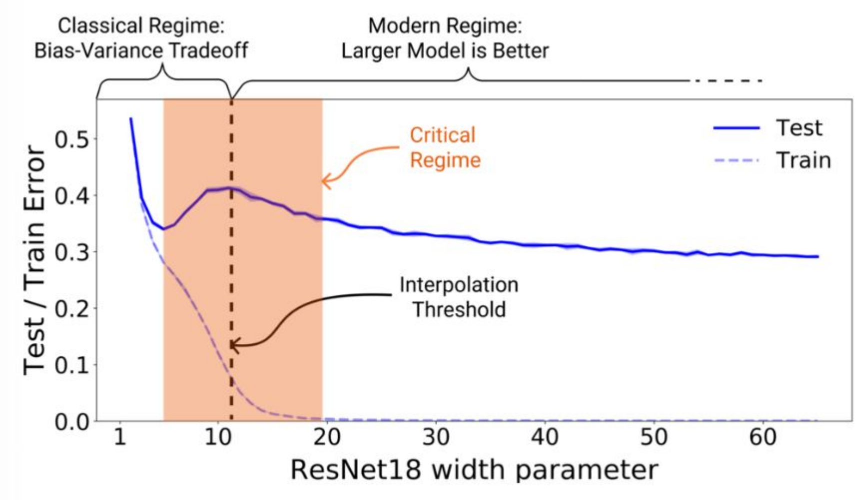
Boaz Barak
Harvard University

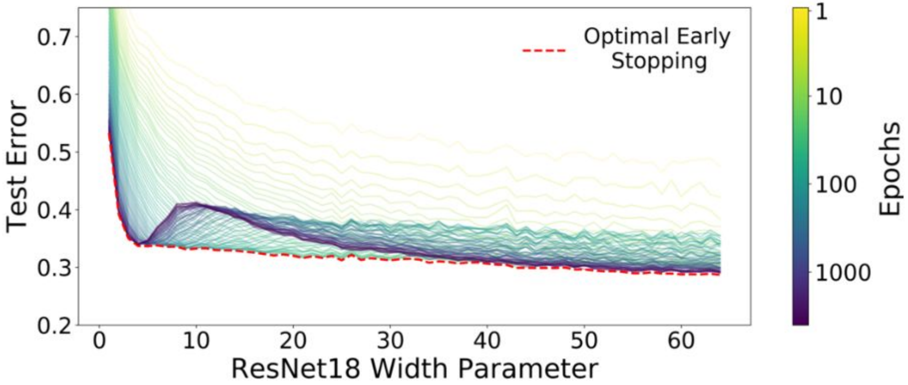
Ilya Sutskever
OpenAI

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.

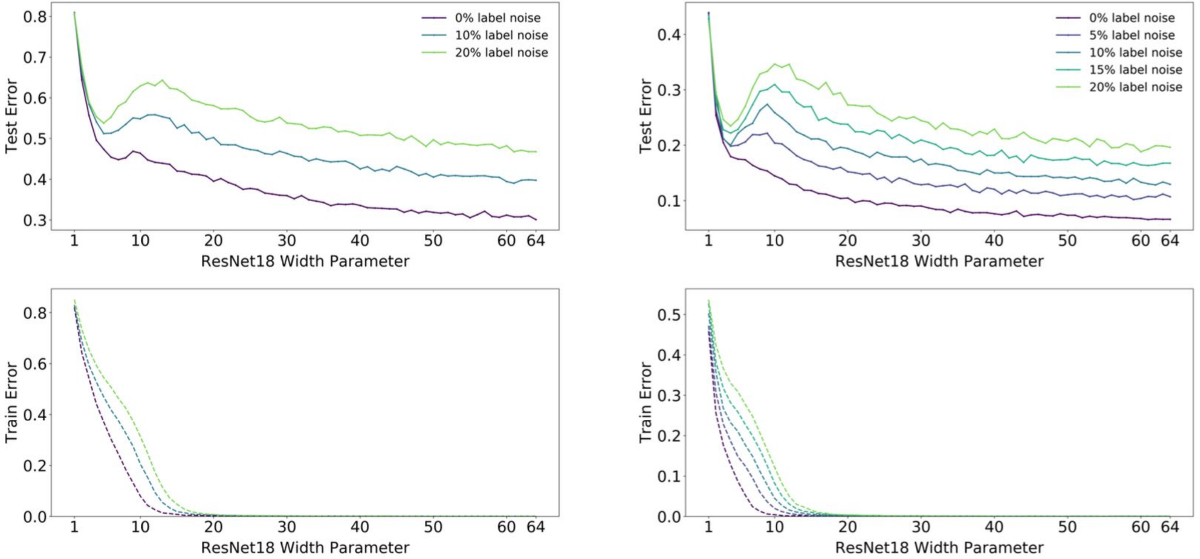
Double Descent





1. Parameters
2. Training Steps
3. Data sizes (n shape)

Model-Wise Double Descent



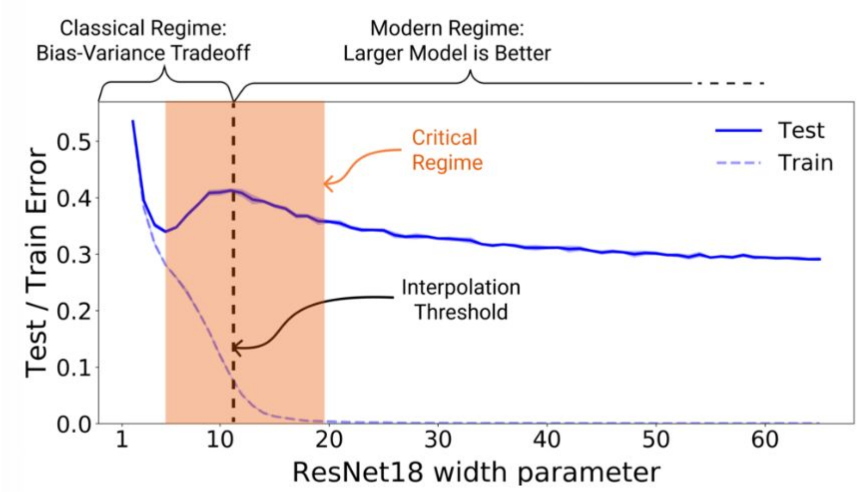
(a) **CIFAR-100.** There is a peak in test error even with no label noise.

(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.

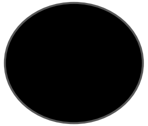
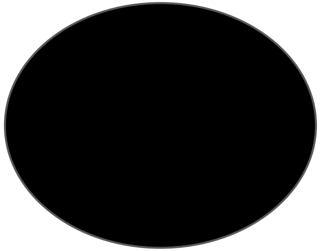
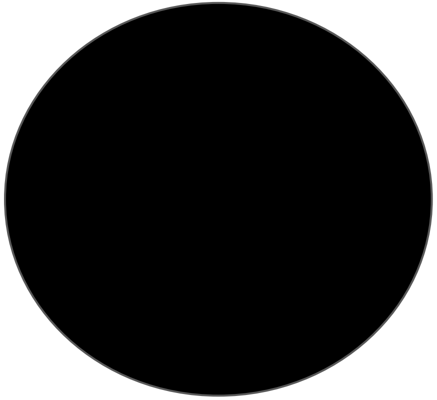
Figure 4: **Model-wise double descent for ResNet18s.** Trained on CIFAR-100 and CIFAR-10, with varying label noise. Optimized using Adam with LR 0.0001 for 4K epochs, and data-augmentation.

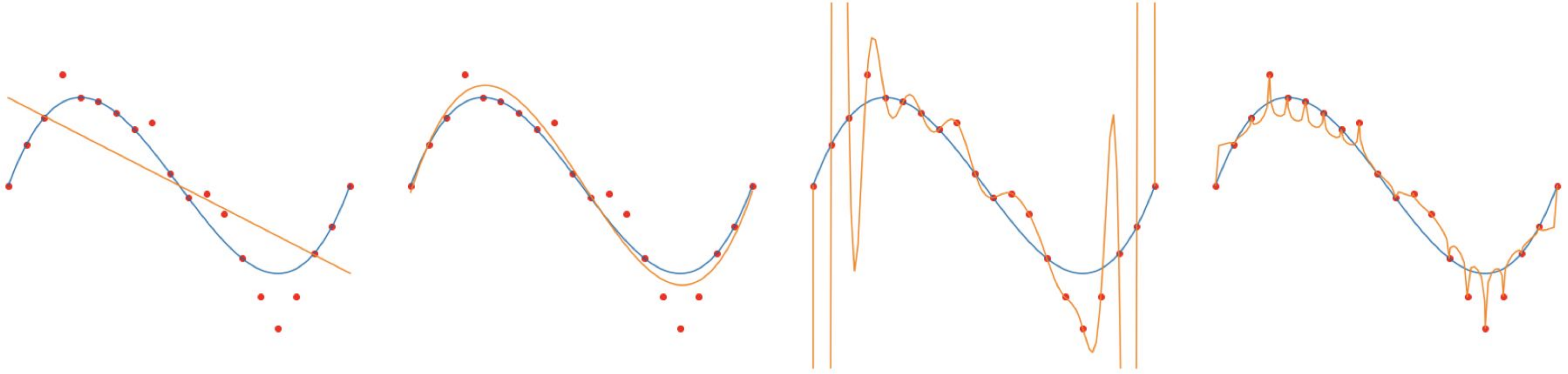
Fixed epochs

Model-Wise Double Descent: Intuition



Bigger set of optima
-> can select a better model



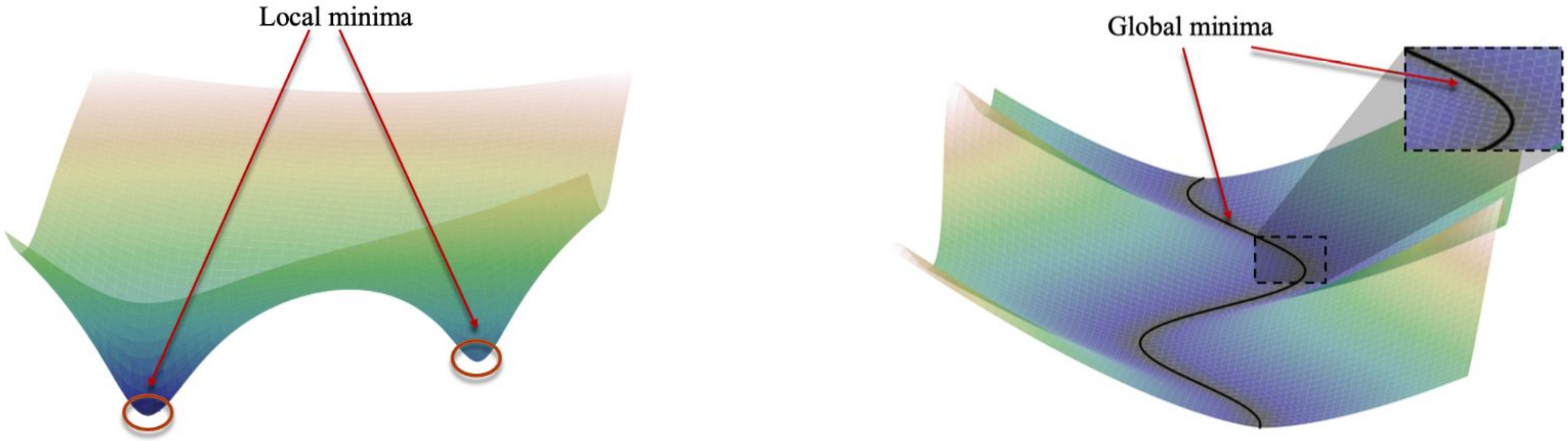
(a) $d = 1$ **(b)** $d = 3$ **(c)** $d = 20$ **(d)** $d = 1000$

Figure 4: Fitting degree d Legendre polynomials (orange curve) to $n = 20$ noisy samples (red dots), from a polynomial of degree 3 (blue curve). Gradient descent is used to minimize the squared error, which leads to the smallest norm solution (considering the norm of the vector of coefficients). Taken from [17].

Lafon & Thomas 2021: 17 is a blogpost of the authors of the main paper

Loss landscapes and optimization in over-parameterized non-linear systems and neural networks

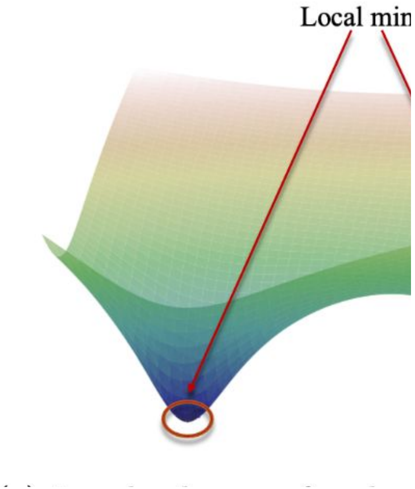
Chaoyue Liu[a], Libin Zhu[b,c], and Mikhail Belkin[c]



(a) Loss landscape of under-parameterized models

(b) Loss landscape of over-parameterized models

Figure 1: Panel (a): Loss landscape is locally convex at local minima. Panel (b): Loss landscape incompatible with local convexity as the set of global minima is not locally linear.



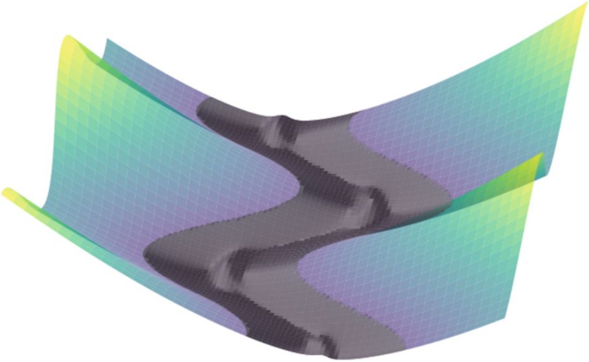
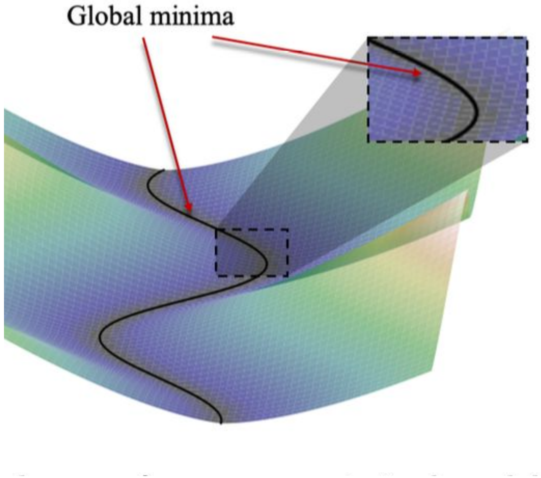


Figure 3: The loss landscape of “almost over-parameterized” systems. The landscape looks over-parameterized except for the grey area where the loss is small. Local minima of the loss are contained there.



Epoch-Wise Double Descent

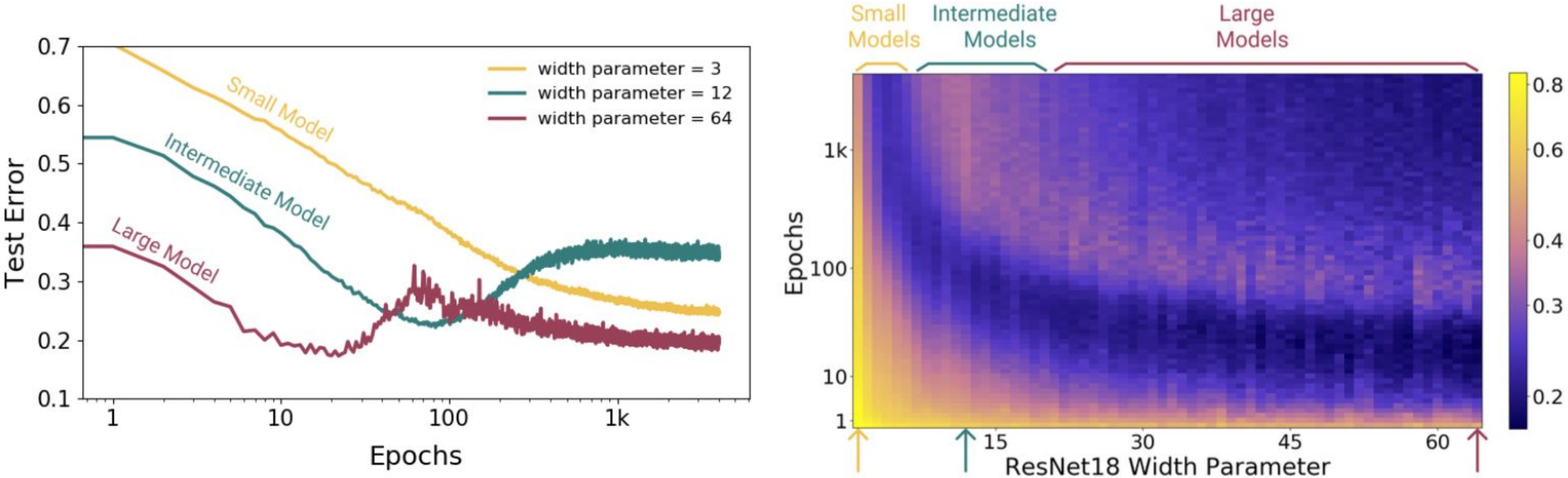


Figure 9: **Left:** Training dynamics for models in three regimes. Models are ResNet18s on CIFAR10 with 20% label noise, trained using Adam with learning rate 0.0001, and data augmentation. **Right:** Test error over (Model size × Epochs). Three slices of this plot are shown on the left.

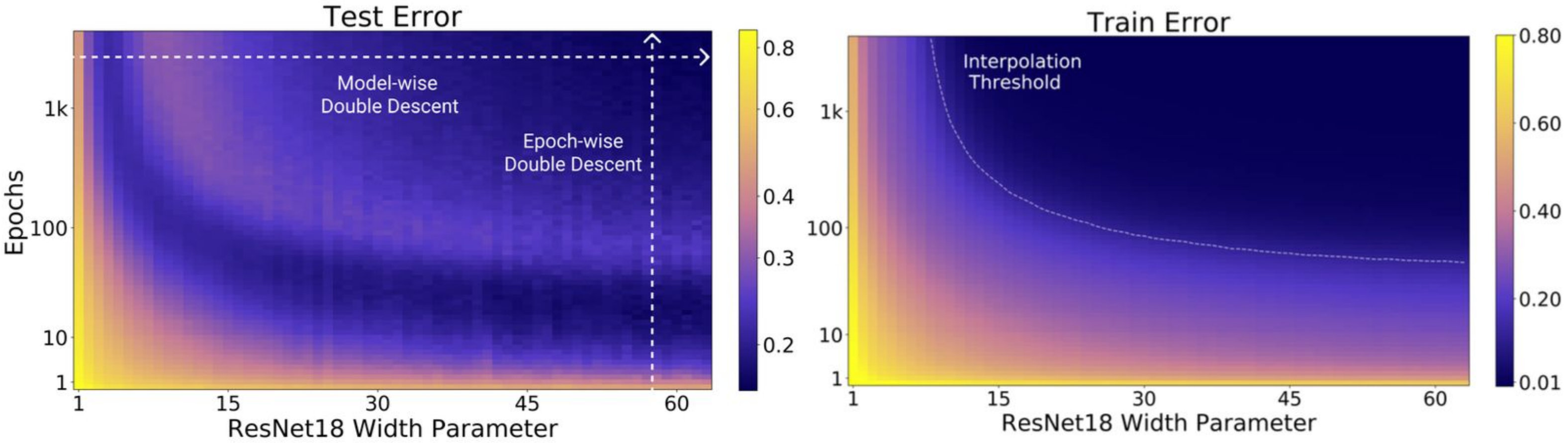
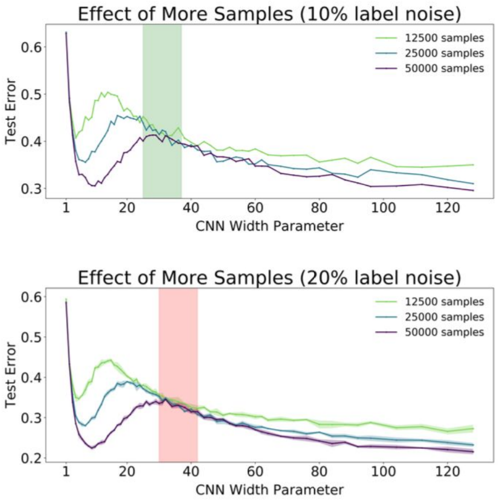


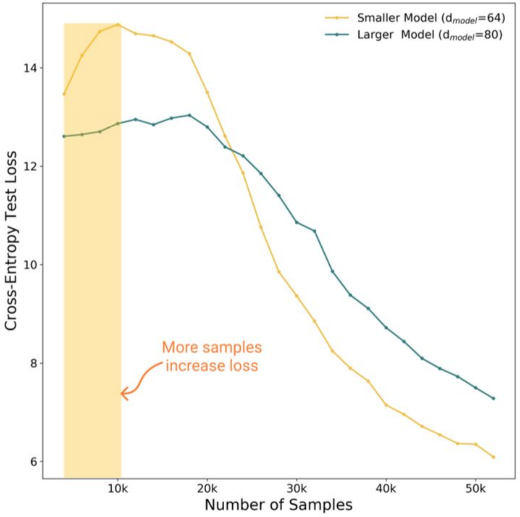
Figure 2: **Left:** Test error as a function of model size and train epochs. The horizontal line corresponds to model-wise double descent–varying model size while training for as long as possible. The vertical line corresponds to epoch-wise double descent, with test error undergoing double-descent as 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.

Discussion: Seems like (almost) any monotone path leads to double descent

Data Non-Monotonicity



(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× more samples does not improve test error. **Bottom:** There is a range of model sizes (shaded red) where training on 4× 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.

Data as Complexity?

Having smaller dataset may essentially reduce expressivity

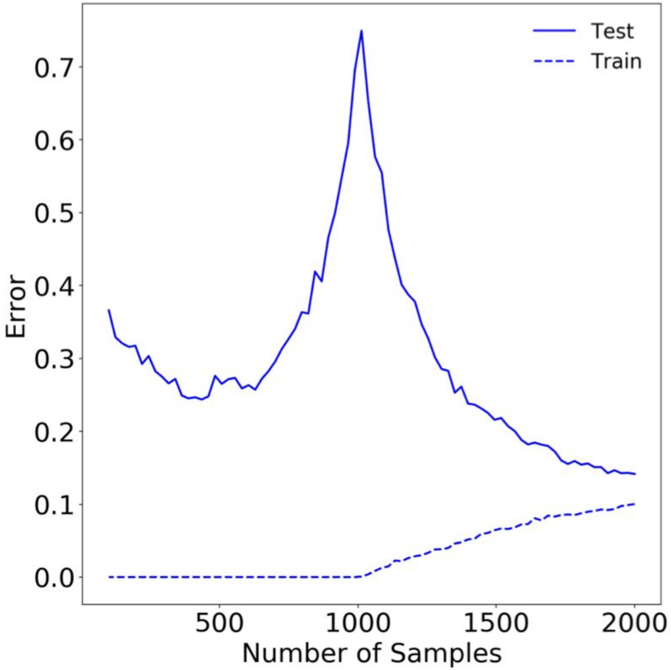
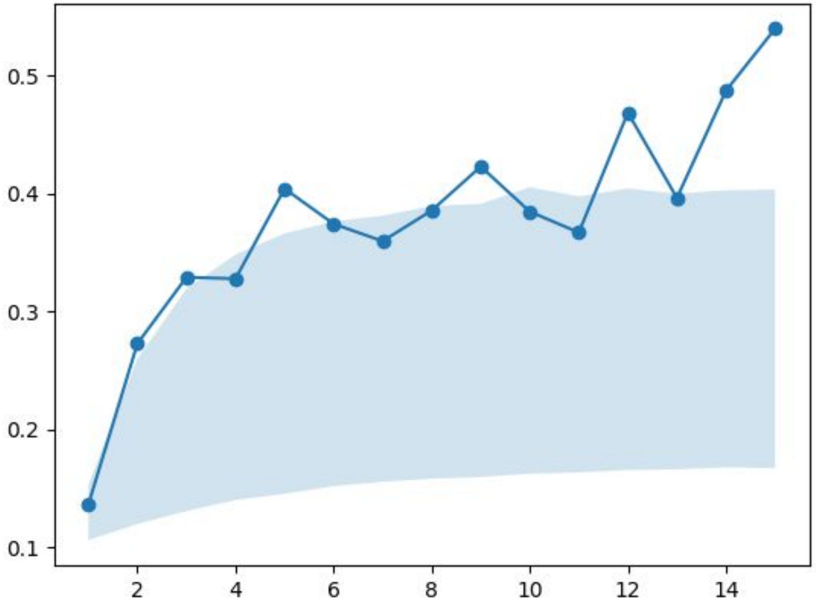
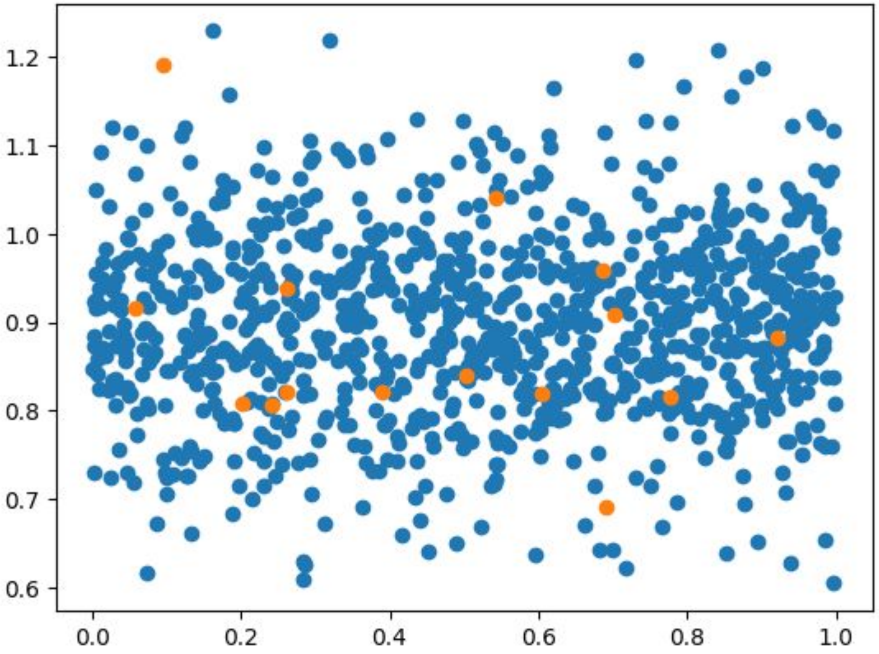


Figure 15: Sample-wise double-descent slice for Random Fourier Features on the Fashion MNIST dataset. In this figure the embedding dimension (number of random features) is 1000.

Toy Example

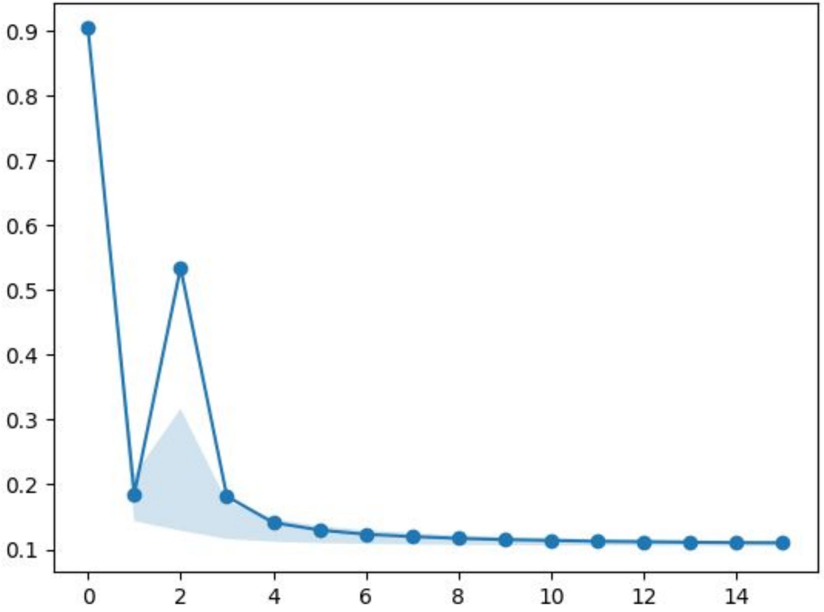
1. Average of 4 Random lines

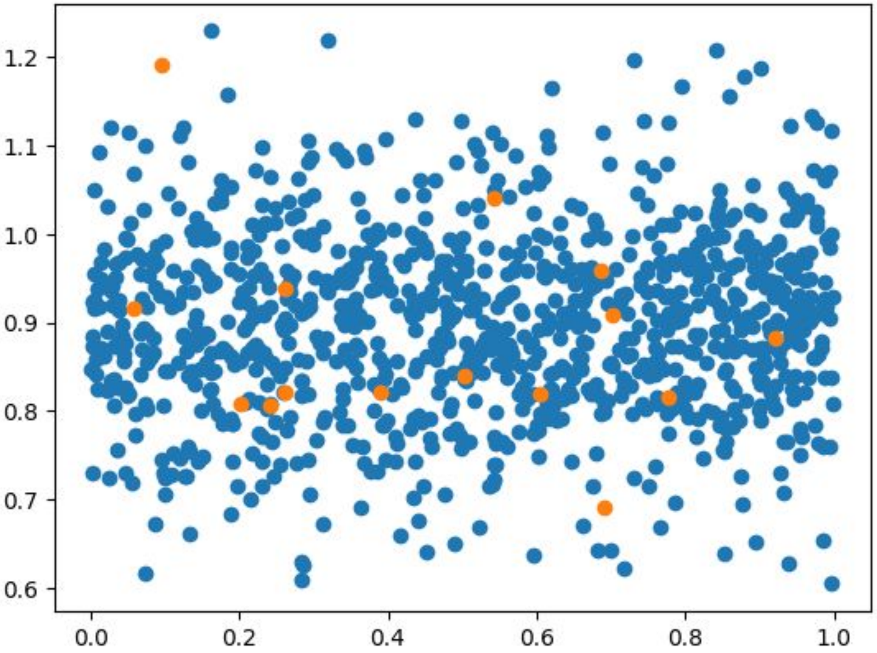




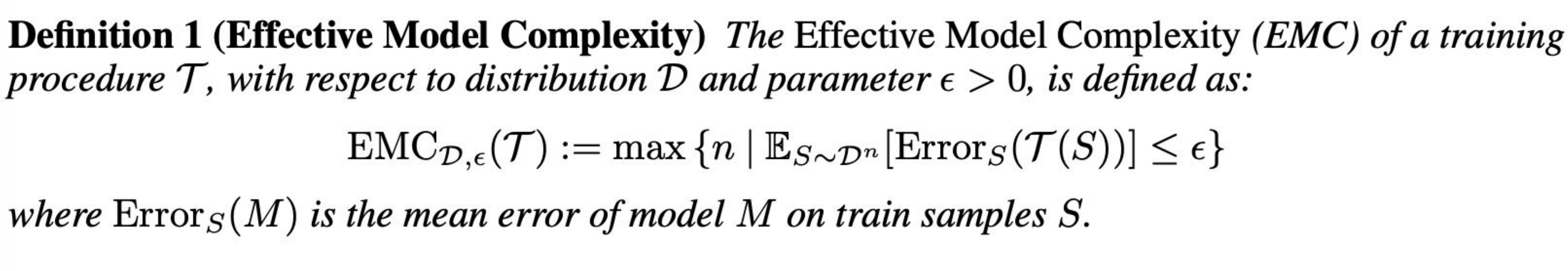
Toy Example: Double Descent?

2. Ridge Regression (with Bias)





Effective Model Complexity (EMC)



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:*

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

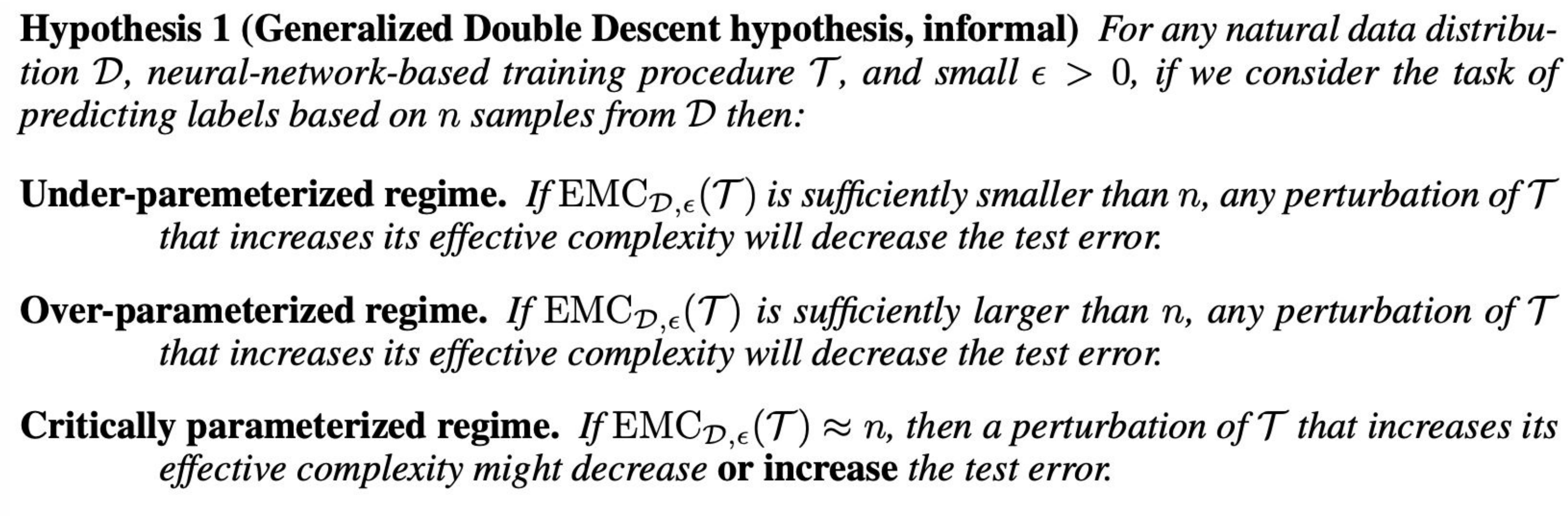
where $\mathrm{Error}_S(M)$ is the mean error of model M on train samples S.

Complexity != Expressivity

Discussion:

- How Training changes with n
- Is the expected training loss monotone?
- Seems possible that there is no such n

Effective Model Complexity (EMC)



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.*