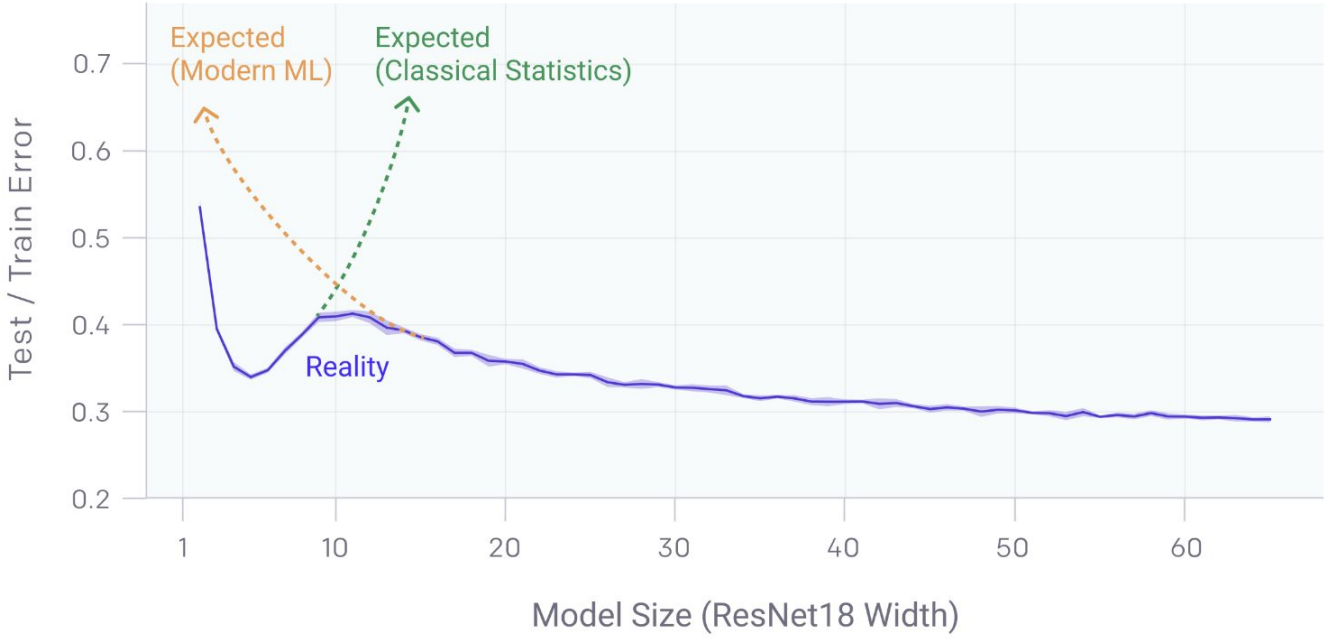
Double descent

Nikita Kozlov, 07/11/2023

Why is it interesting? What does phenomen challenge?

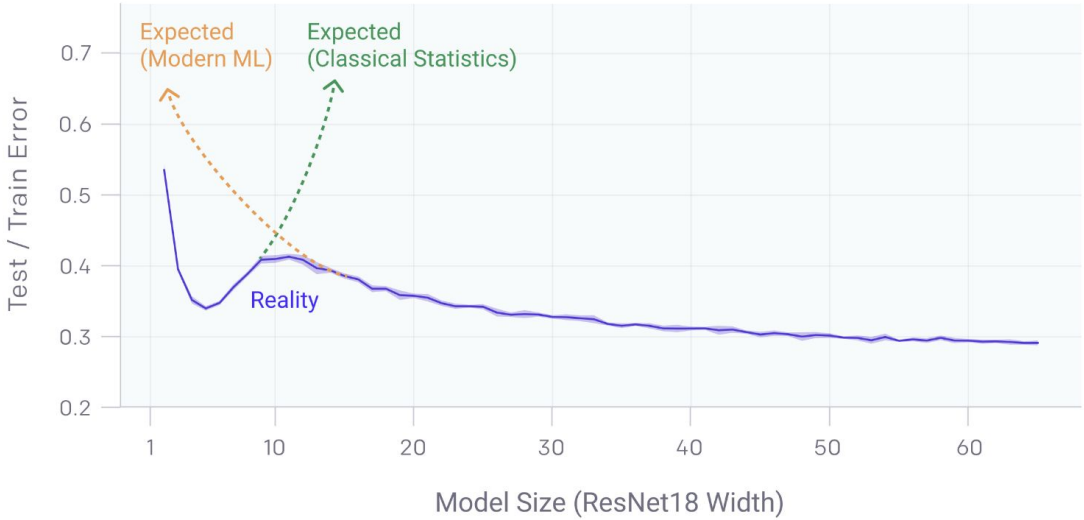
- More data is better
- Larger models are better
- Early stopping

Overview: Bias-Variance Tradeoff

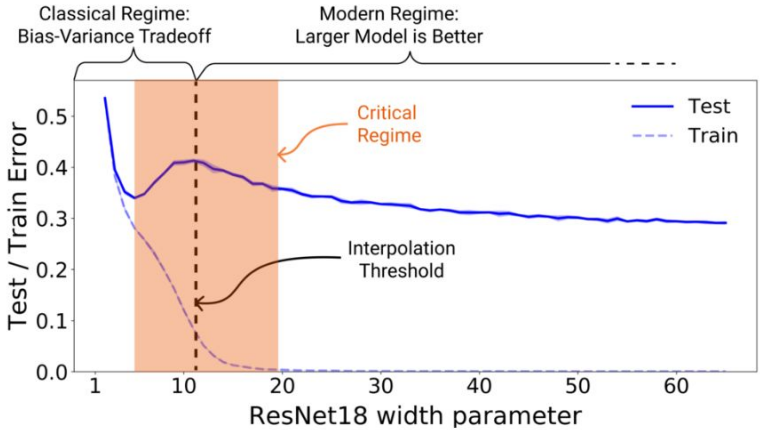


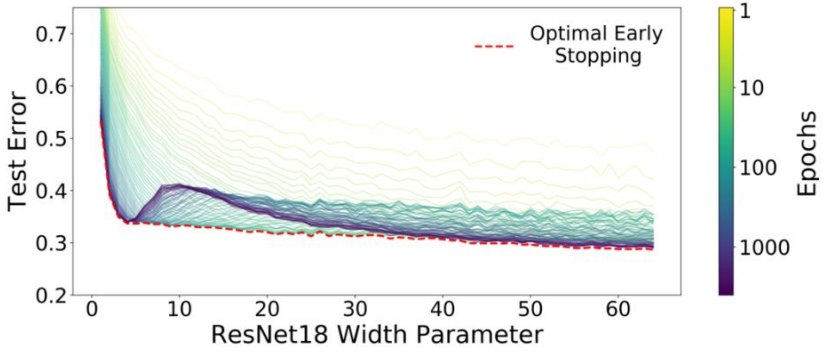
Variations

- Model-wise Double descent: varying # of parameters
- Epoch-wise Double descent: length of training
- Sample-wise non-monotonicity: # of train samples
- Label noise: amount of label noise in distribution



Interpolation Threshold





We reach threshold when model is large enough to fit all the train data.

CIFAR-10 with 15% label noise + Adam

Effective model complexity

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:*

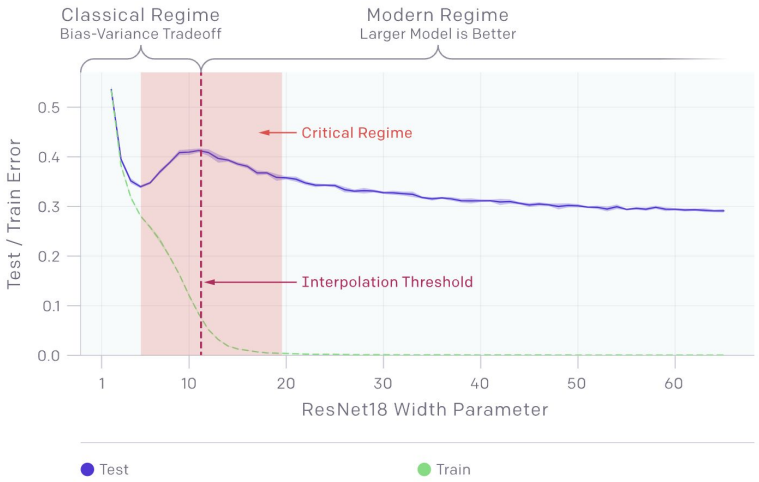
$$\mathrm{EMC}_{\mathcal{D},\epsilon}(\mathcal{T}) := \max\{n \mid \mathbb{E}_{S\sim\mathcal{D}^n}[\mathrm{Error}_S(\mathcal{T}(S))] \leq \epsilon\}$$

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

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.*

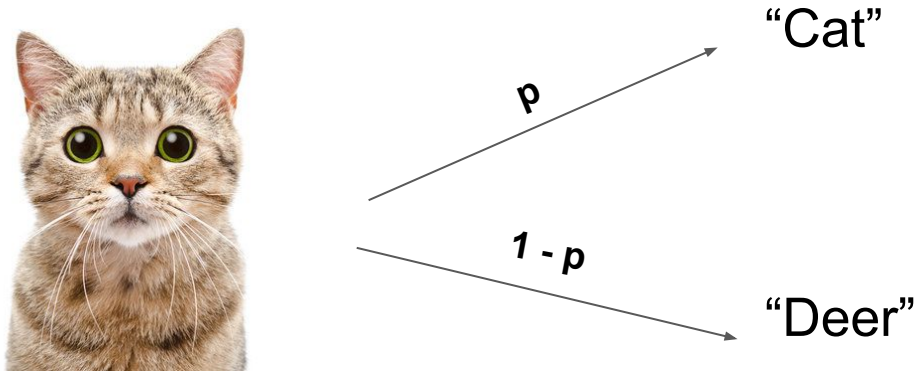


TL;DR: EMC = # of samples the model with current capacity can “model”

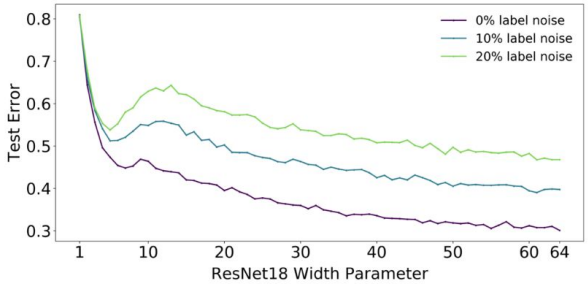
Label noise introduction

With probability **p** we would have the correct label, and with probability **1 - p** we would choose random label from uniform distribution of labels.

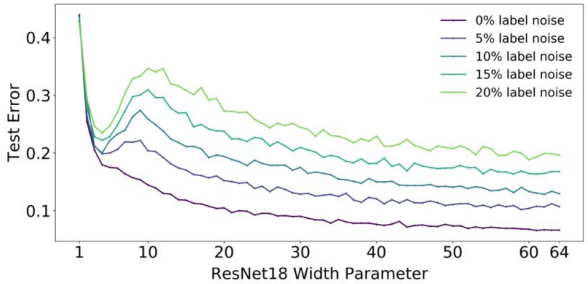
*Label noise generated only once at the beginning of training.



Model-wise + label noise



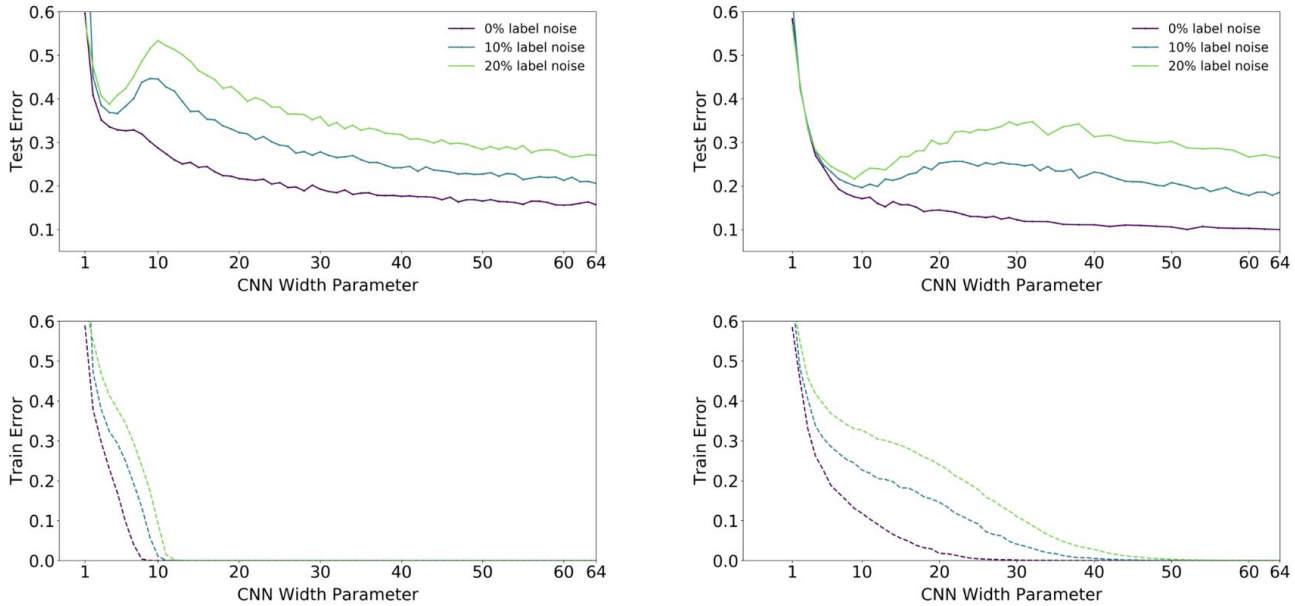
(a) **CIFAR-100.** There is a peak in test error even with no label noise.



(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.

Data Augmentation Impact



(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.

Epoch-wise

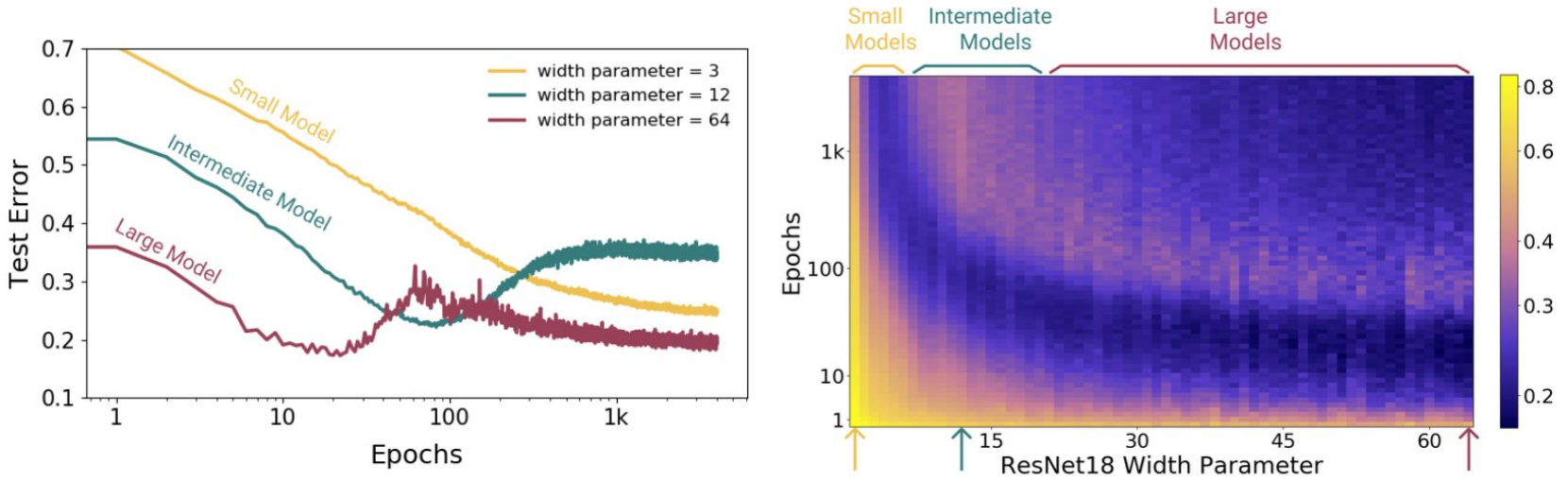
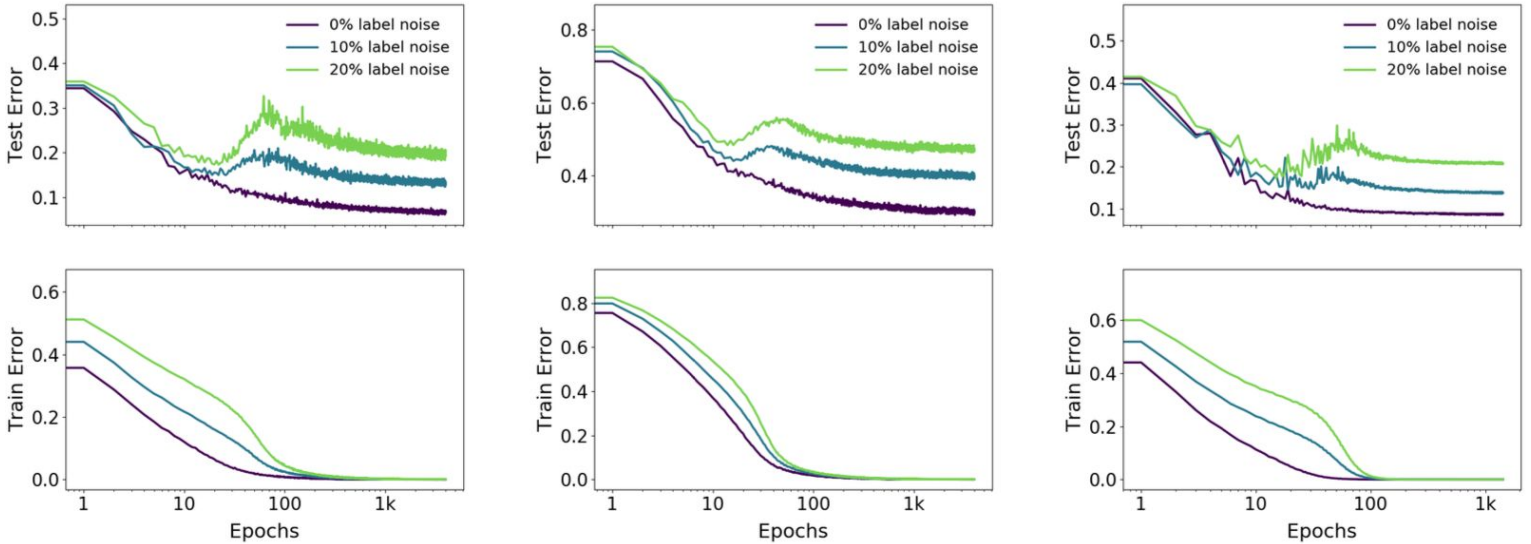


Figure 9: **Left:** Training dynamics for models in three regimes. Models are ResNet18s on CIFAR10 with 20% label noise, trained using Adam with learning rate 0.0001, and data augmentation. **Right:** Test error over (Model size $\times$ Epochs). Three slices of this plot are shown on the left.

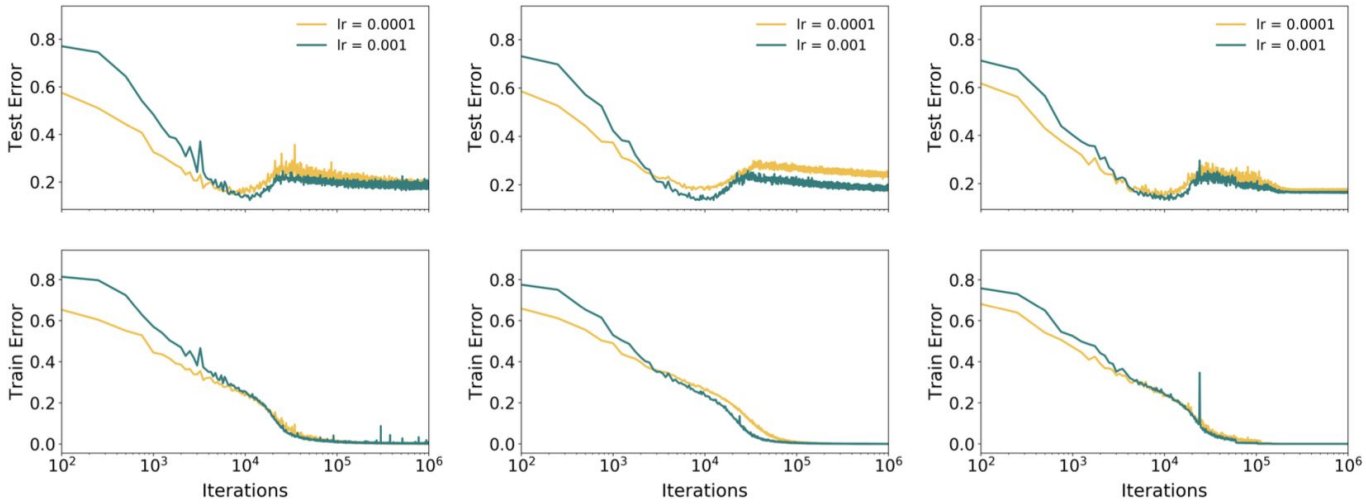
Epoch-wise + label noise



(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.

Epoch-wise + Adam + lr

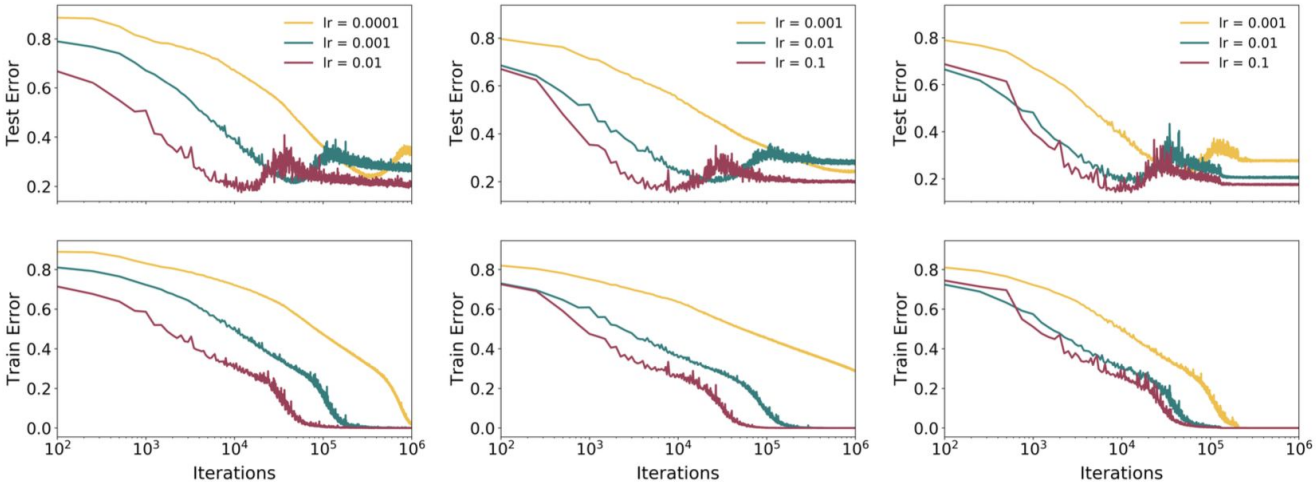


(a) Constant learning rate (b) Inverse-square root learning rate (c) Dynamic learning rate

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

ResNet18 on CIFAR-10 with data-augmentation and 20% label noise

Epoch-wise + SGD + lr



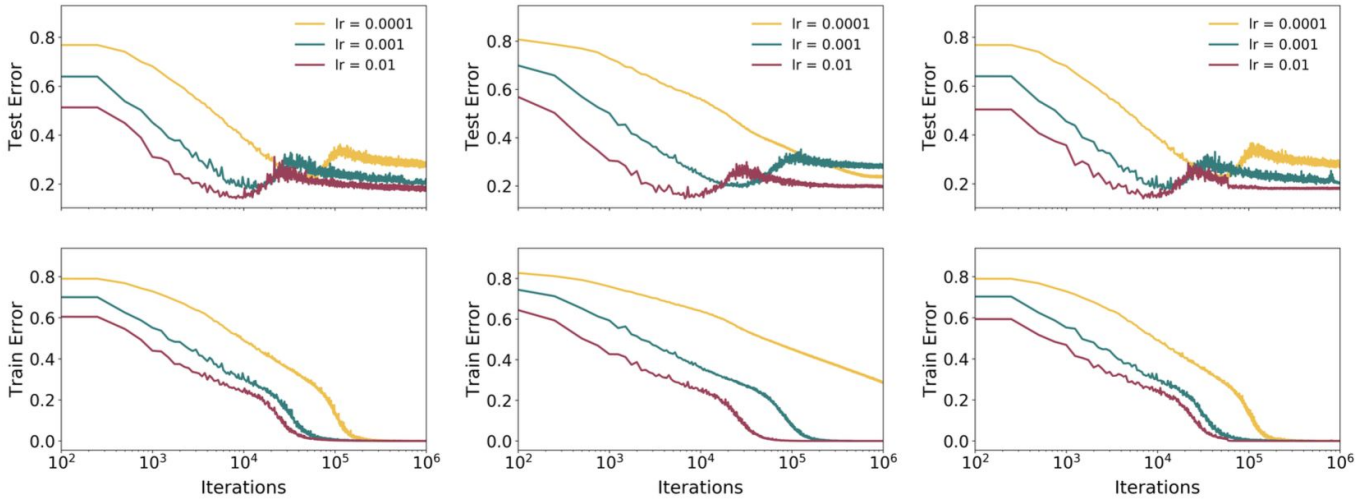
(a) Constant learning rate

(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

Epoch-wise + SGD+Momentum + lr



(a) Constant learning rate

(b) Inverse-square root learning rate

(c) Dynamic learning rate

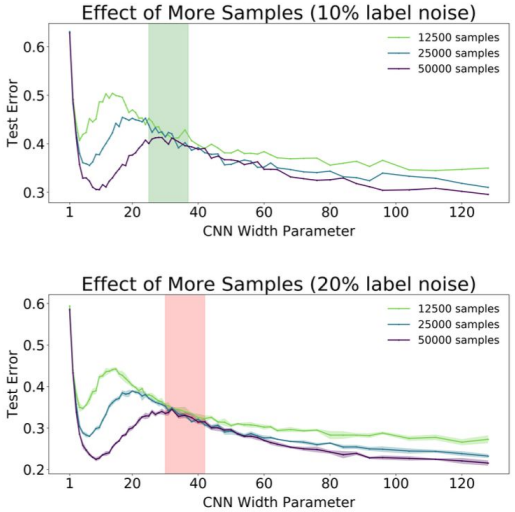
Figure 18: **Epoch-wise double descent** for ResNet18 trained with SGD+Momentum and multiple learning rate schedules

Momentum set to 0.9

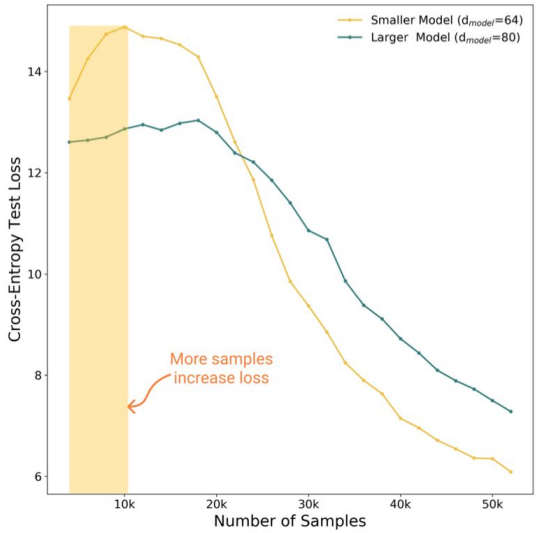
Sample-wise non-monotonicity

IWSLT‘14 German-to-English

Sample-wise non-monotonicity + label noise



(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.

Figure 11: Sample-wise non-monotonicity.

Possible explanations

Relation with Lottery Ticket Hypothesis

- We are choosing the model with minimum norm in increasing spaces (because there is increasing number of degrees of freedom).
- “SGD seeks out and trains a well-initialized subnetwork” - (Reconciling modern machine learning practice and the bias-variance trade-off)
- Choosing the smoothest function that perfectly fits observed data is a form of Occam’s razor. By considering larger function classes, which contain more candidate predictors compatible with the data, we are able to find interpolating functions that have smaller norm and are thus “simpler”.
- In wide linear model fit by least squares, SGD with a small step size leads to a minimum norm zero-residual solution.

More real-world examples

Language models

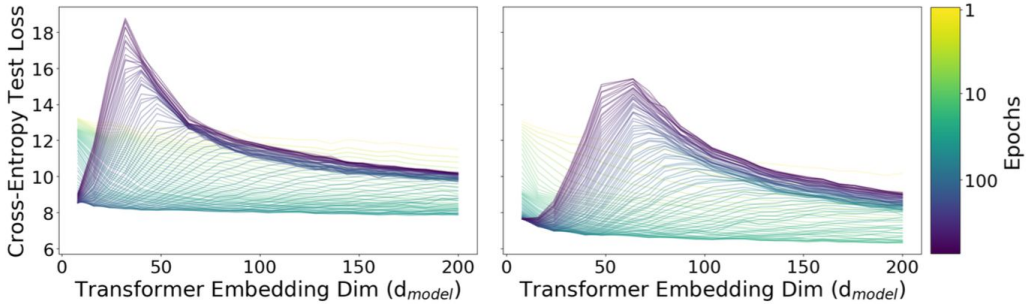


Figure 23: Model-wise test error dynamics for a subsampled IWSLT‘14 dataset. Left: 4k samples, Right: 18k samples. Note that with optimal early-stopping, more samples is always better.

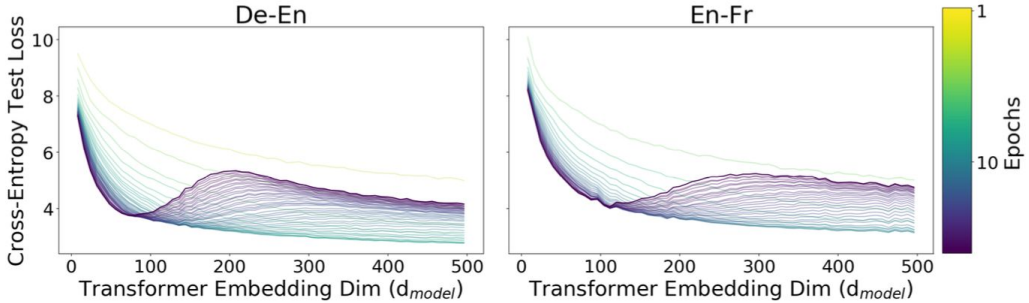


Figure 24: Model-wise test error dynamics for a IWSLT‘14 de-en and subsampled WMT‘14 en-fr datasets. **Left**: IWSLT‘14, **Right**: subsampled (200k samples) WMT‘14. Note that with optimal early-stopping, the test error is much lower for this task.

Ensembling Impact

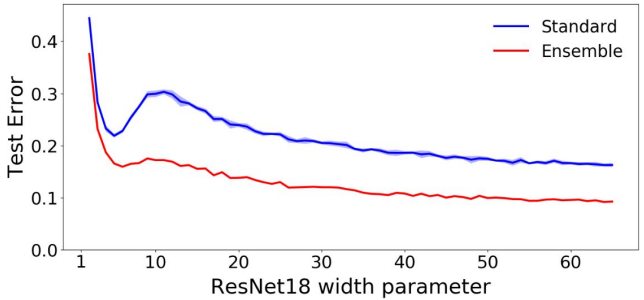


Figure 28: **Effect of Ensembling (ResNets, 15% label noise).** Test error of an ensemble of 5 models, compared to the base models. The ensembled classifier is determined by plurality vote over the 5 base models. Note that emsembling helps most around the critical regime. All models are ResNet18s trained 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.

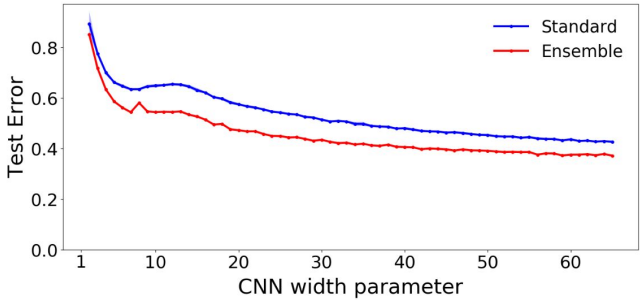


Figure 29: **Effect of Ensembling (CNNs, no label noise).** Test error of an ensemble of 5 models, compared to the base models. All models are 5-layer CNNs trained on CIFAR-10 with no label noise, using SGD and no data augmentation. (same setting as Figure 7).

References

- Deep Double Descent: Where Bigger Models and More Data Hurt
- OpenAI Blog Post: Deep Double Descent
- Deep Double Descent explanation video
- Reconciling modern machine learning practice and the bias-variance trade-off
- Statistical Learning: 10.7 Interpolation and Double Descent from Stanford Online
- Ilya Sutskever on Deep Double Descent