基于非校验集信息的深度学习泛化性能曲线预测

伍冬睿

华中科技大学

drwu@hust.edu.cn

Content

1. Generalization, Memorization, and Spectral Bias of DNNs

2. Optimization Variance (OV)

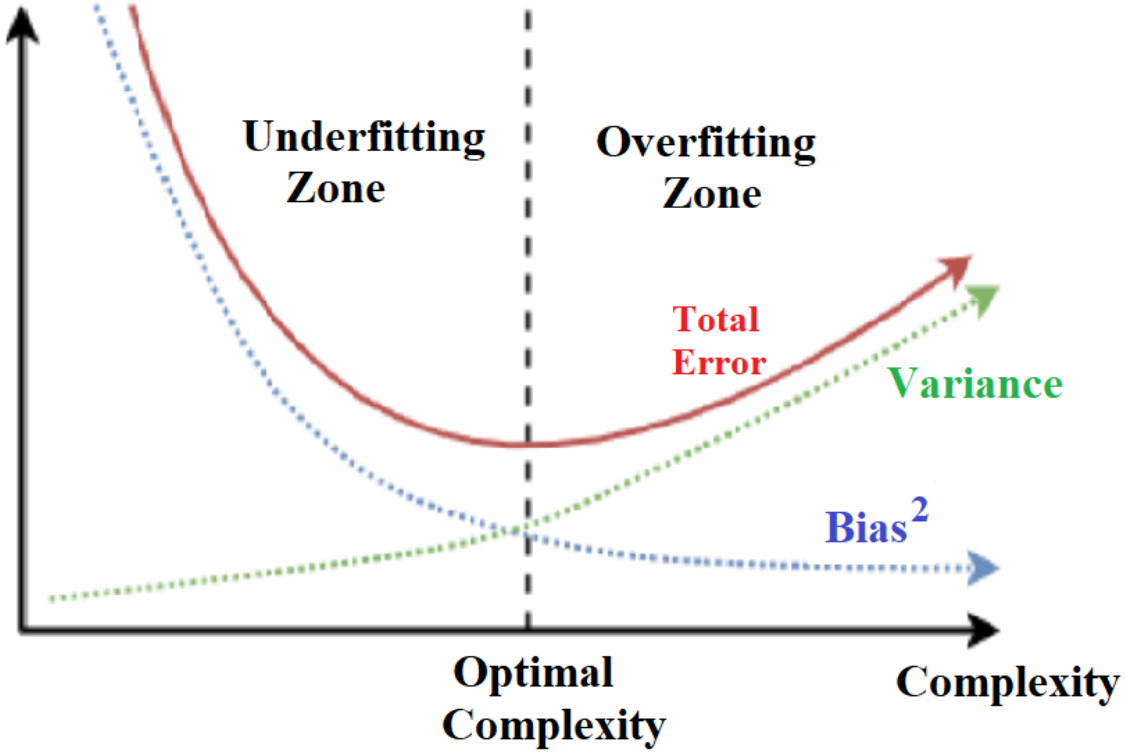
Content

1. **Generalization, Memorization, and Spectral Bias of DNNs**

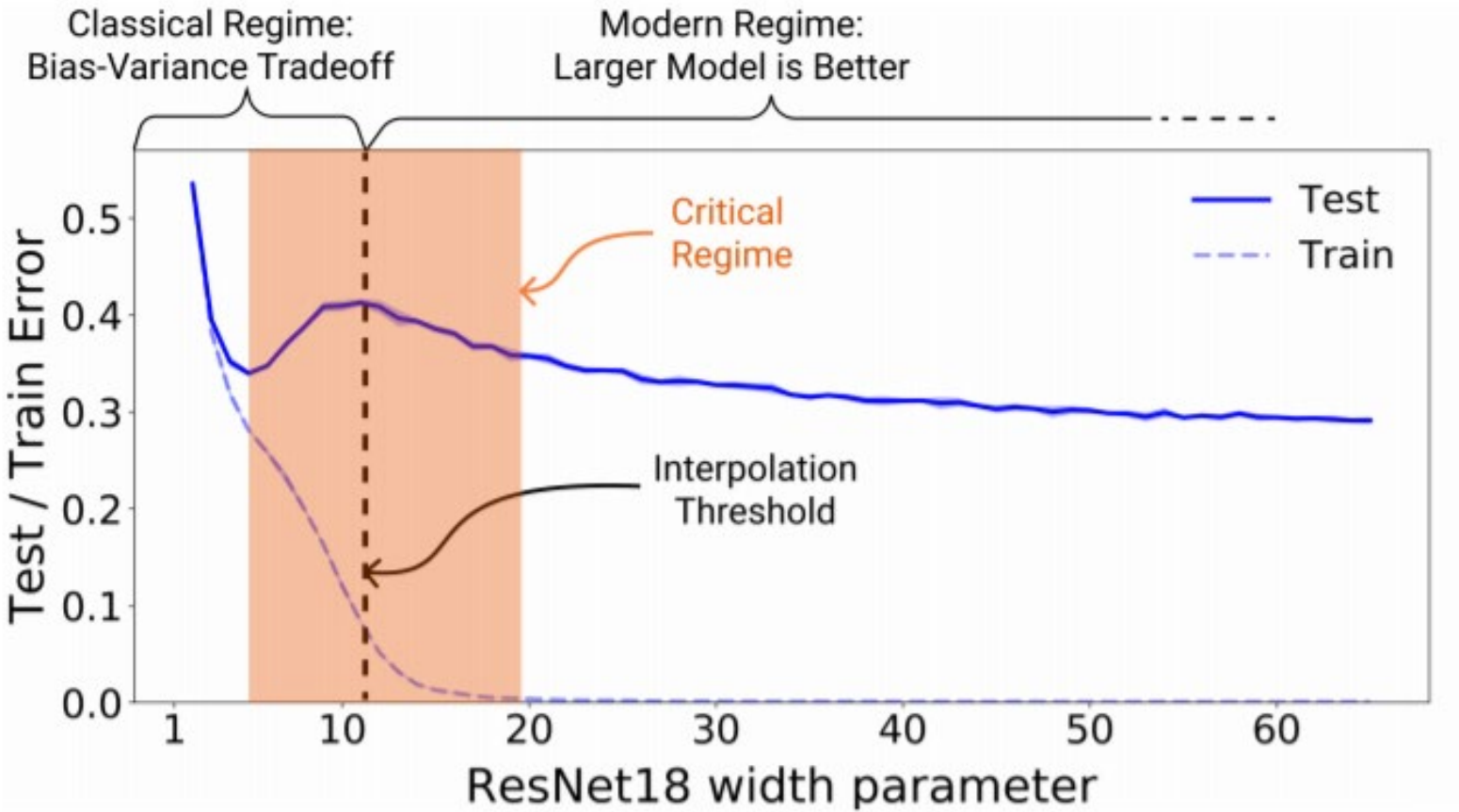
2. Optimization Variance (OV)

Reference: X. Zhang, H. Xiong and D. Wu, "Rethink the Connections among Generalization, Memorization, and the Spectral Bias of DNNs," IJCAI 2021.

Traditional Bias-Variance Trade-off



Bias-Variance Trade-off in DNNs



Reference: P. Nakkiran et al., Deep double descent: Where bigger models and more data hurt. ICLR 2020

Generalization of DNNs

- Over-parameterized networks have powerful expressivity to completely memorize all training examples with random labels, yet they can still generalize well on normal examples

- Explicit regularization may improve generalization, but is neither necessary nor by itself sufficient for controlling generalization error

  ✓ **Explicit regularization**: dropout, weight-decay, data augmentation

  ✓ **Implicit regularization**: early stopping, batch normalization, **stochastic gradient descent (SGD)**

Reference: C. Zhang, et al., Understanding deep learning requires rethinking generalization. ICLR 2017

The Role of SGD in DNN Training

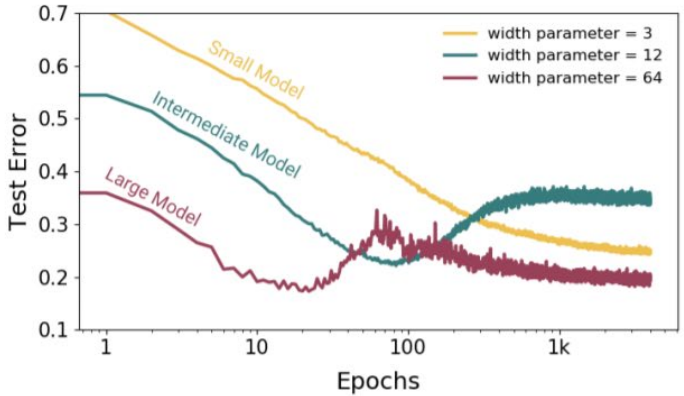
SGD has an inductive bias to search the hypothesises which show excellent generalization performances:

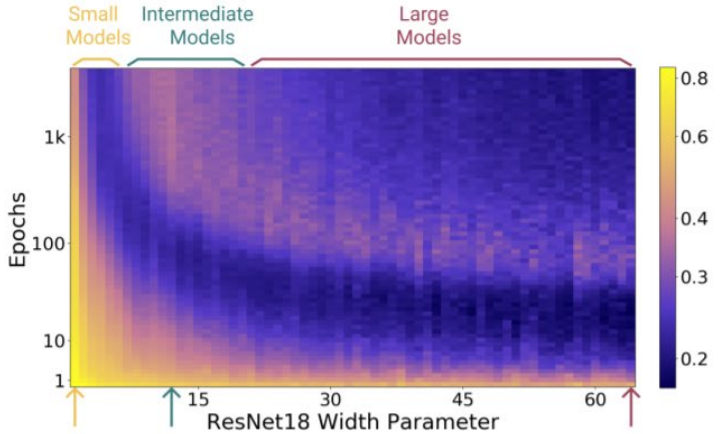
- DNNs learn patterns first, and then use brute-force memorization to fit the noise hard to generalize [1]
- SGD on DNNs learns functions of increasing complexity gradually [2]
- **Spectral bias** (frequency principle) of DNNs [3,4]: Lower frequencies in the input space are learned first and then the higher ones. Overfitting happens when the complexity of models keeps increasing or high-frequency components remain being introduced

References:
[1] D. Arpit et al., A closer look at memorization in deep networks. ICML 2017
[2] D. Kalimeris et al., SGD on neural networks learns functions of increasing complexity. NeurIPS 2019
[3] N. Rahaman et al., On the spectral bias of neural networks. ICML 2019
[4] Z-Q Xu et al., Training behavior of deep neural network in frequency domain, NeurIPS 2019

Epoch-wise Double Descent

- **Previous finding**: The learning bias in training DNNs is monotonic, e.g., from simple to complex or from low frequencies to high frequencies
- **Conflicting** with epoch-wise double descent [1] : The generalization error first has a classical U-shaped curve and then follows a second descent
- According to spectral bias, with higher-frequency components being gradually introduced in training, the generalization performance should deteriorate monotonically due to the memorization of noise



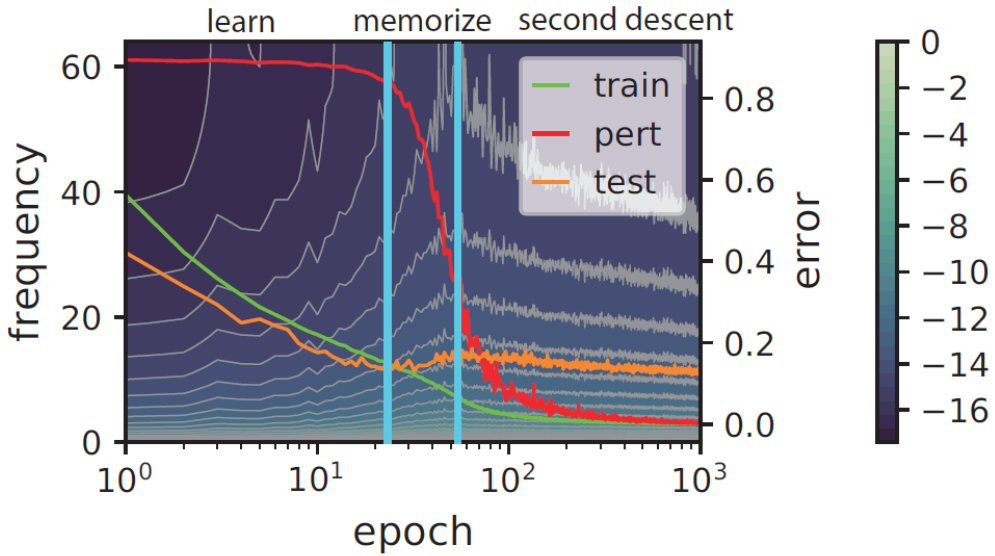


Reference: [1] P. Nakkiran et al., Deep double descent: Where bigger models and more data hurt. ICLR 2020

Our Main Findings

- At a certain epoch, usually around the start of the 2nd descent, while the perturbed part is still being memorized, the high-frequency components begin to diminish

- **Reason**: The prediction surface off the training data manifold becomes flatter and more regularized in the late training stage, which improves the generalization performance of models on the test samples not covered by the training data manifold. As a result, the second descent of the test error happens.



Reference: X. Zhang, H. Xiong and D. Wu, "Rethink the Connections among Generalization, Memorization, and the Spectral Bias of DNNs," IJCAI 2021.

Spectrum of a DNN

Denote the input point sampled from a distribution $\mathcal{D}$ by $\boldsymbol{x} \sim \mathcal{D}$, a normalized random direction by $\boldsymbol{v}_x$, the c-th logit output of a DNN by $f_c(\boldsymbol{x})$, where $c \in \{1, 2, ..., C\}$ and C is the number of classes.

We evenly sample N points from $[\boldsymbol{x} - h\boldsymbol{v}_x, \boldsymbol{x} + h\boldsymbol{v}_x]$ to perform the discrete Fourier transform, where h bounds the area.

The Fourier transform of $f_c(\boldsymbol{x})$ is then:

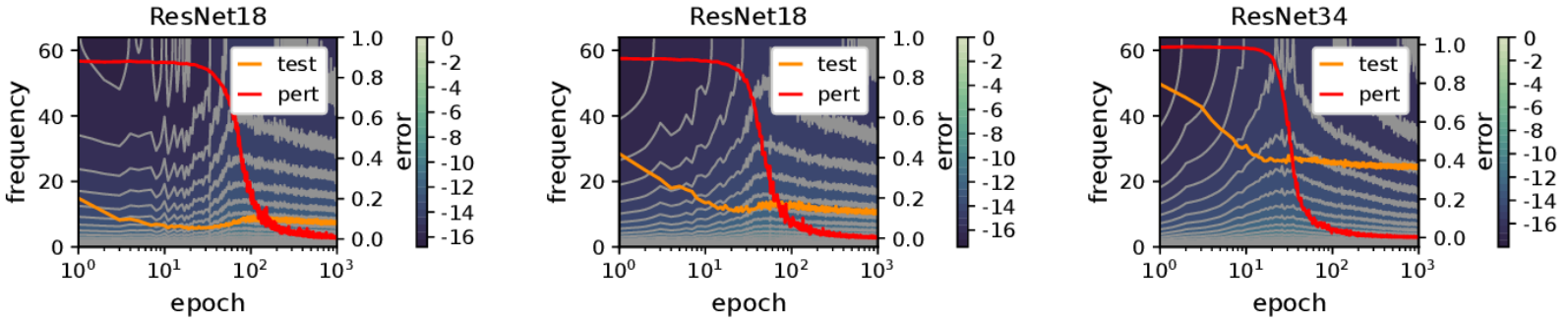
$$\tilde{f}_{c,\boldsymbol{x}}(k) = \sum_{n=1}^{N} f_c\left(\boldsymbol{x} + \frac{2n - N - 1}{N - 1} h\boldsymbol{v}_x\right) e^{-i2\pi \frac{n}{N} k}.$$

We then add up the spectra across the dataset and the logit outputs to illustrate the local variation from a global viewpoint:

$$A_k = \frac{1}{C} \sum_{c=1}^{C} \mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}} \left| \tilde{f}_{c,\boldsymbol{x}}(k) \right|^2.$$

Epoch-wise Double Descent

Perturbed set: Randomly shuffled 10% labels of the training set to strengthen double descent

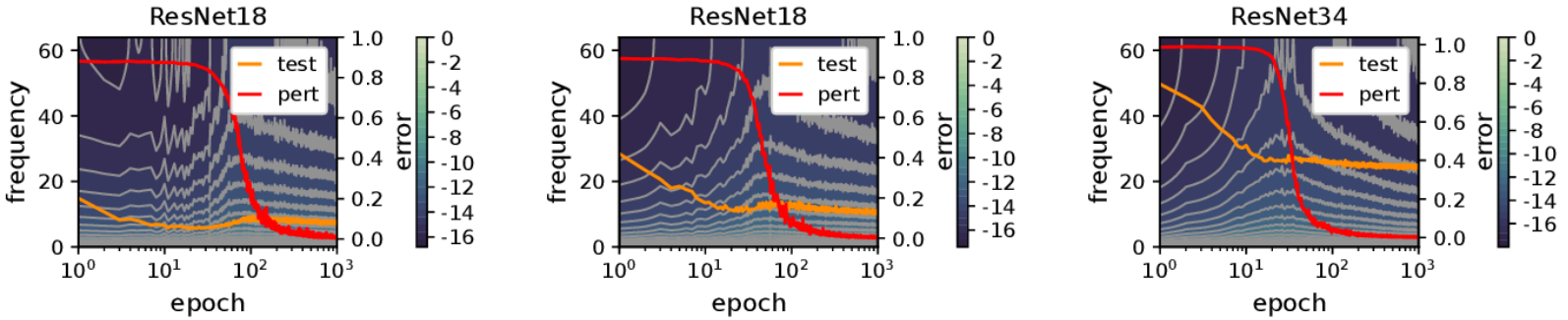


(a) SVHN (b) CIFAR10 (c) CIFAR100

- The test error decreases quickly at the beginning of the training, whereas the error on the perturbed set remains high: the models effectively learn the patterns of the data in this period
- As the training goes on, the error on the perturbed set decreases rapidly, along with a little increase of the test error. In this period, the models start to memorize the noise, which leads to overfitting on the training set
- **Epoch-wise double descent**: If the models are trained with more epochs, the peak of the test error occurs, just around the epoch when the model memorizes the noise, and then the test error steps into the second descent

Non-Monotonicity of the Spectral Bias

Perturbed set: Randomly shuffled 10% labels of the training set to strengthen double descent



(a) SVHN (b) CIFAR10 (c) CIFAR100

- The models first introduce low-frequency components and then the high-frequency ones
- The ratio of the high-frequency components increases rapidly when the models try to memorize the noise
- Along the 2nd descent of the test error, the high-frequency components begin to diminish, violating the claims about the monotonicity of the spectral bias

Why Do High-Frequency Components Diminish?

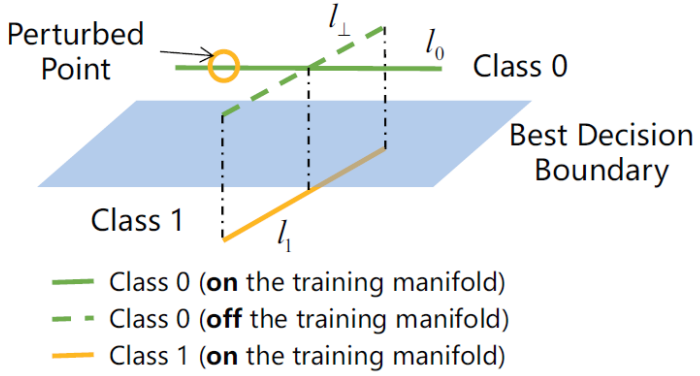
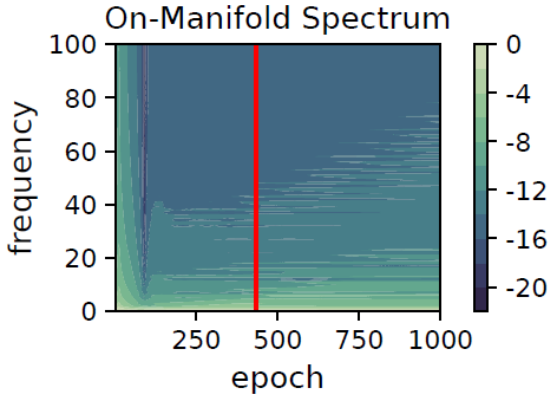
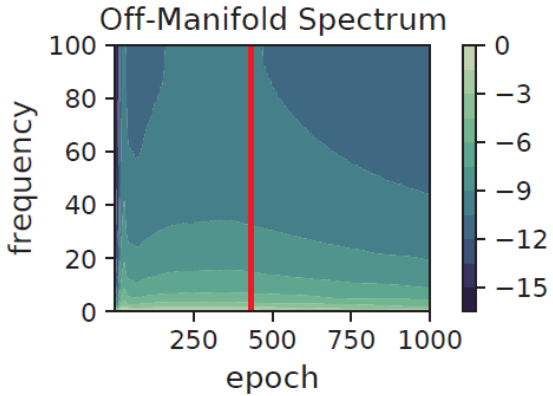
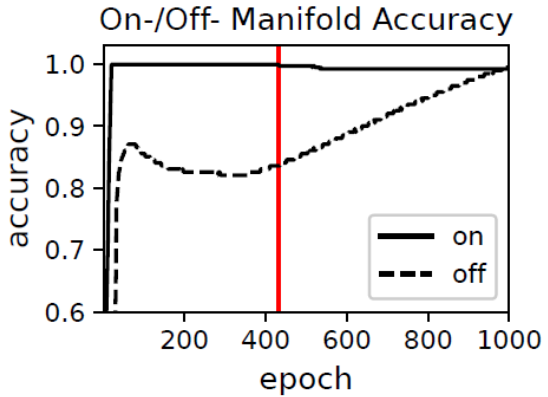


Figure 3: Illustration of the toy task. The blue plane is the desired decision boundary separating l_0 and l_1. The perturbed point is labeled as Class 1. The off-manifold line $l_\perp$ intersects with the on-manifold line l_0 perpendicularly. The task is to train the model on l_0 and l_1, and observe the spectra on l_0 and $l_\perp$.

- The on-manifold spectrum keeps introducing high-frequency components to memorize the perturbed point
- After the perturbed point is memorized, the off-manifold spectrum is more biased towards the low-frequency components, resulting in a much flatter off-manifold prediction surface
- The on-manifold accuracy slightly decreases after memorizing the perturbed point, whereas the off-manifold accuracy keeps increasing







Summary

- We studied the frequency components of DNNs in the data point neighbors via Fourier analysis
- The monotonicity of the spectral bias does not always hold, because the off-manifold prediction surface may reduce its high-frequency components in the late training stage
- Though perturbed points on the training data manifold remain memorized by the on-manifold prediction surface, this implicit regularization on the off-manifold prediction surface can still help improve the generalization performance

Reference: X. Zhang, H. Xiong and D. Wu, "Rethink the Connections among Generalization, Memorization, and the Spectral Bias of DNNs," IJCAI 2021.

1. Generalization, Memorization, and Spectral Bias of DNNs

2. **Optimization Variance (OV)**

Reference: X. Zhang, D. Wu*, H. Xiong and B. Dai, “Optimization Variance: Exploring Generalization Properties of DNNs," *https://arxiv.org/abs/2106.01714*

Main Contributions

- We perform bias-variance decomposition on the test error to explore epoch-wise double descent.
- We show that for the 0/1 loss, the variance highly correlates with the variation of the test error
- We propose optimization variance (OV), which is calculated from the training set only and correlates well with the test error
- Based on the OV, we propose an approach to search for the early stopping point without using a validation set, when the 0/1 loss is used in test

Unified Bias-Variance Decomposition

Let $(\boldsymbol{x}, \boldsymbol{t})$ be a sample drawn from the data distribution $\mathcal{D}$, where $\boldsymbol{x} \in \mathbb{R}^d$ denotes the d-dimensional input, and $\boldsymbol{t} \in \mathbb{R}^c$ the one-hot encoding of the label in c classes. The training set $\mathcal{T} = \{(\boldsymbol{x}_i, \boldsymbol{t}_i)\}_{i=1}^n \sim \mathcal{D}^n$ is utilized to train the model $f : \mathbb{R}^d \to \mathbb{R}^c$. Let $\boldsymbol{y} = f(\boldsymbol{x}; \mathcal{T}) \in \mathbb{R}^c$ be the probability output of the model f trained on $\mathcal{T}$, and $\mathcal{L}(\boldsymbol{t}, \boldsymbol{y})$ the loss function.

A unified bias-variance decomposition of $\mathbb{E}_{\mathcal{T}}[\mathcal{L}(\boldsymbol{t}, \boldsymbol{y})]$ is:

$$\mathbb{E}_{\mathcal{T}}[\mathcal{L}(\boldsymbol{t}, \boldsymbol{y})] = \underbrace{\mathcal{L}(\boldsymbol{t}, \bar{\boldsymbol{y}})}_{\text{Bias}} + \beta \underbrace{\mathbb{E}_{\mathcal{T}}[\mathcal{L}(\bar{\boldsymbol{y}}, \boldsymbol{y})]}_{\text{Variance}},$$

where β takes different values for different loss functions, and $\bar{\boldsymbol{y}}$ is the expected output:

$$\bar{\boldsymbol{y}} = \underset{\boldsymbol{y}^* \in \mathbb{R}^c \,\middle|\, \sum_{k=1}^c \boldsymbol{y}_k^* = 1, \boldsymbol{y}_k^* \geq 0}{\arg\min} \mathbb{E}_{\mathcal{T}}[\mathcal{L}(\boldsymbol{y}^*, \boldsymbol{y})].$$

$\bar{\boldsymbol{y}}$ minimizes the variance term, which can be regarded as the "center" or "ensemble" of $\boldsymbol{y}$ w.r.t. different $\mathcal{T}$.

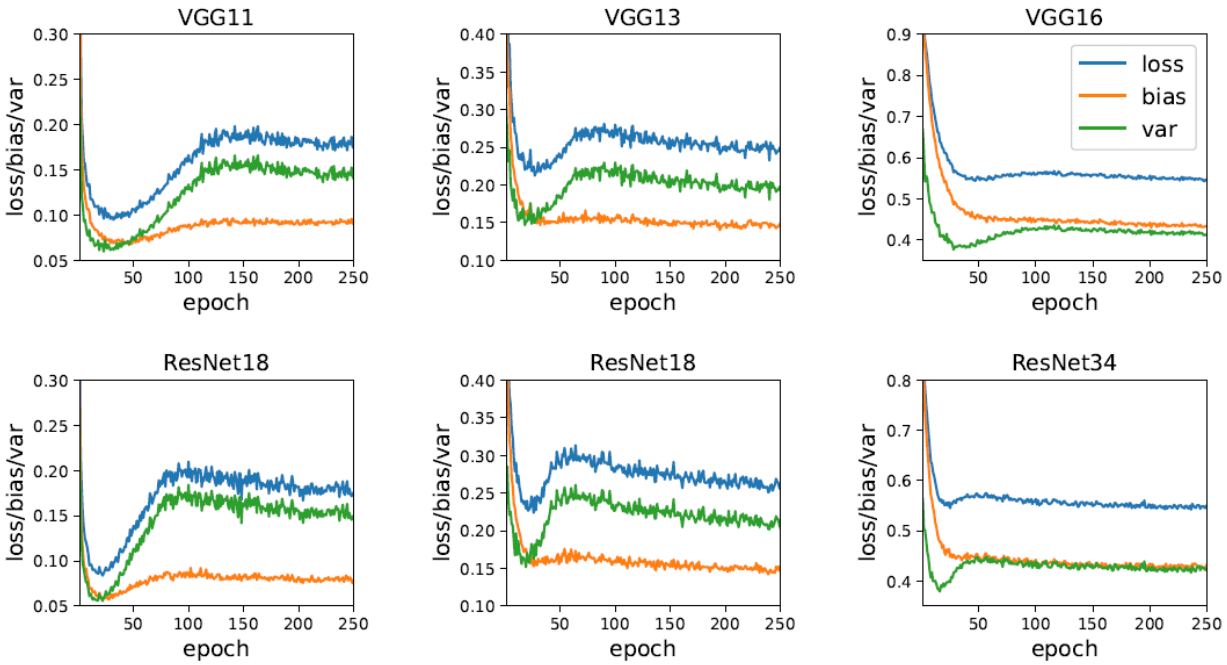
Unified Bias-Variance Decomposition

Table 1: Bias-variance decomposition for different loss functions. The CE loss herein is the complete form of the commonly used one, originated from the Kullback-Leibler divergence. $Z = \sum_{k=1}^{c} \exp\{\mathbb{E}_{\mathcal{T}}[\log y_k]\}$ is a normalization constant independent of k. $\mathrm{H}(\cdot)$ is the hard-max which sets the maximal element to 1 and others to 0. $\mathbf{1}_{\mathrm{con}}\{\cdot\}$ is an indicator function which equals 1 if its argument is true, and 0 otherwise. log and exp are element-wise operators.

Loss	$\mathcal{L}(\boldsymbol{t}, \boldsymbol{y})$	$\bar{\boldsymbol{y}}$	β
MSE	$\|\boldsymbol{t} - \boldsymbol{y}\|_2^2$	$\mathbb{E}_{\mathcal{T}}\boldsymbol{y}$	1
CE	$\sum_{k=1}^{c} t_k \log \frac{t_k}{y_k}$	$\frac{1}{Z}\exp\{\mathbb{E}_{\mathcal{T}}[\log \boldsymbol{y}]\}$	1
ZO	$\mathbf{1}_{\mathrm{con}}\{\mathrm{H}(\boldsymbol{t}) \neq \mathrm{H}(\boldsymbol{y})\}$	$\mathrm{H}(\mathbb{E}_{\mathcal{T}}[\mathrm{H}(\boldsymbol{y})])$	1 if $\bar{\boldsymbol{y}} = \boldsymbol{t}$, otherwise $-P_{\mathcal{T}}(\mathrm{H}(\boldsymbol{y}) = \boldsymbol{t} \mid \bar{\boldsymbol{y}} \neq \mathrm{H}(\boldsymbol{y}))$

Bias-Variance Decomposition on the Test Set

- The bias descends rapidly at first and then generally converges to a low value
- The variance behaves almost exactly the same as the test error, mimicking even small fluctuations of the test error
- **Mainly the variance contributes to epoch-wise double descent**



(a) SVHN (b) CIFAR10 (c) CIFAR100

The Variance

- The variance measures the model diversity caused by different training samples drawn from the same distribution, i.e., the outputs of DNN change according to the sampled training set
- As the gradients are usually the only information transferred from training sets to models during the optimization of DNN, we measure the variance of a DNN introduced by the gradients calculated from different training batches
- More specifically, we'd like to develop a metric to reflect the function robustness of DNNs to sampling noise
- If the function captured by a DNN drastically varies w.r.t. different training batches, then very likely it has poor generalization due to a large variance introduced by the optimization procedure

Optimization Variance (OV)

For a sample $(\boldsymbol{x}, \boldsymbol{t}) \sim \mathcal{D}$, let $f(\boldsymbol{x}; \boldsymbol{\theta})$ be the logit output of a DNN with parameter $\boldsymbol{\theta}$.

Let $\mathcal{T}_B \sim \mathcal{D}^m$ be a training batch with m samples, $g : \mathcal{T}_B \to \mathbb{R}^{|\boldsymbol{\theta}|}$ the optimizer outputting the update of $\boldsymbol{\theta}$ based on $\mathcal{T}_B$.

The function distribution $F_{\boldsymbol{x}}(\mathcal{T}_B)$ over a training batch $\mathcal{T}_B$ can be approximated as $f(\boldsymbol{x}; \boldsymbol{\theta} + g(\mathcal{T}_B))$.

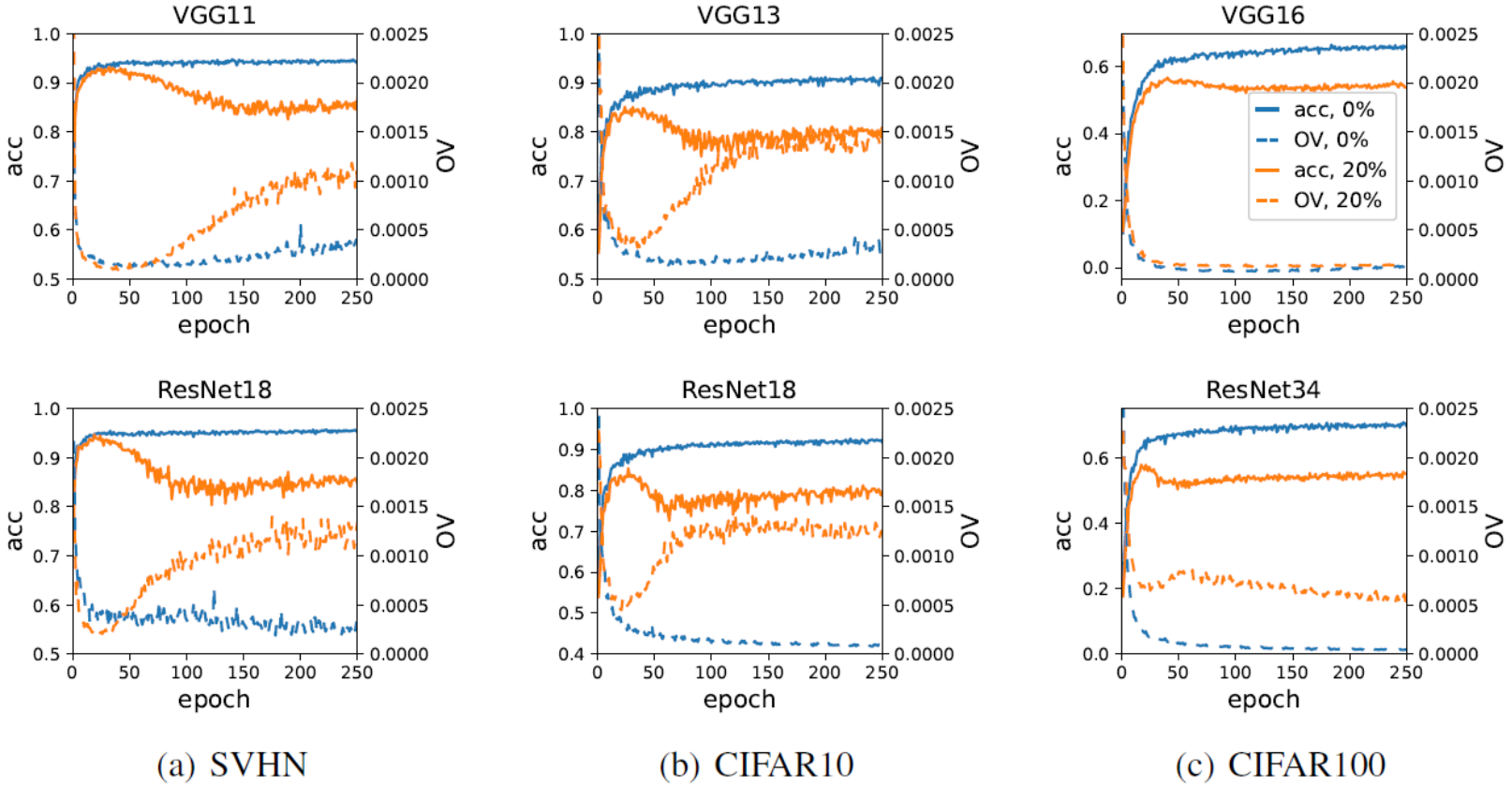
The variance of $F_{\boldsymbol{x}}(\mathcal{T}_B)$ reflects the model diversity caused by different training batches.

Given an input $\boldsymbol{x}$ and model parameters $\boldsymbol{\theta}_q$ at the q-th training epoch, the OV on $\boldsymbol{x}$ at the q-th epoch is defined as

$$OV_q(\boldsymbol{x}) \triangleq \frac{\mathbb{E}_{\mathcal{T}_B}\left[\left\|f(\boldsymbol{x}; \boldsymbol{\theta}_q + g(\mathcal{T}_B)) - \mathbb{E}_{\mathcal{T}_B} f(\boldsymbol{x}; \boldsymbol{\theta}_q + g(\mathcal{T}_B))\right\|_2^2\right]}{\mathbb{E}_{\mathcal{T}_B}\left[\left\|f(\boldsymbol{x}; \boldsymbol{\theta}_q + g(\mathcal{T}_B))\right\|_2^2\right]}.$$

If $OV_q(\boldsymbol{x})$ is very large, the models trained with different $\mathcal{T}_B$ may have distinct outputs for the same input, leading to high model diversity and hence large variance.

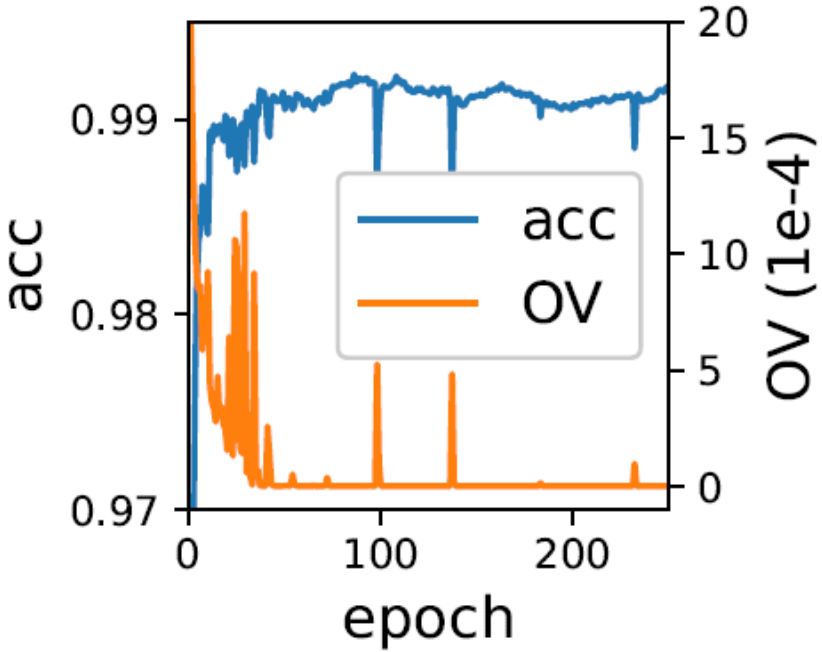
Characteristics of OV



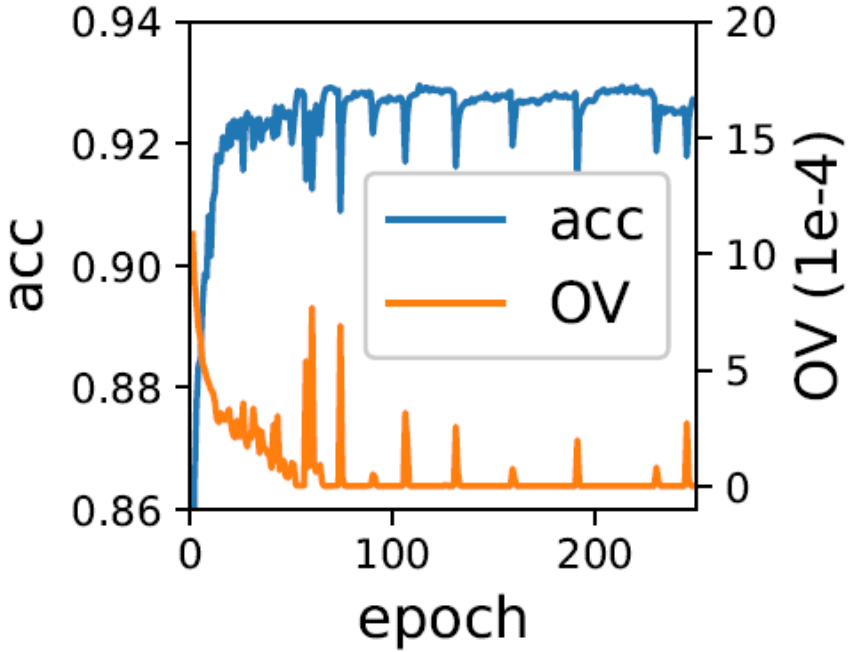
(a) SVHN (b) CIFAR10 (c) CIFAR100

OV, which is calculated **from the training set only**, is capable of predicting the variation of the test accuracy

Characteristics of OV



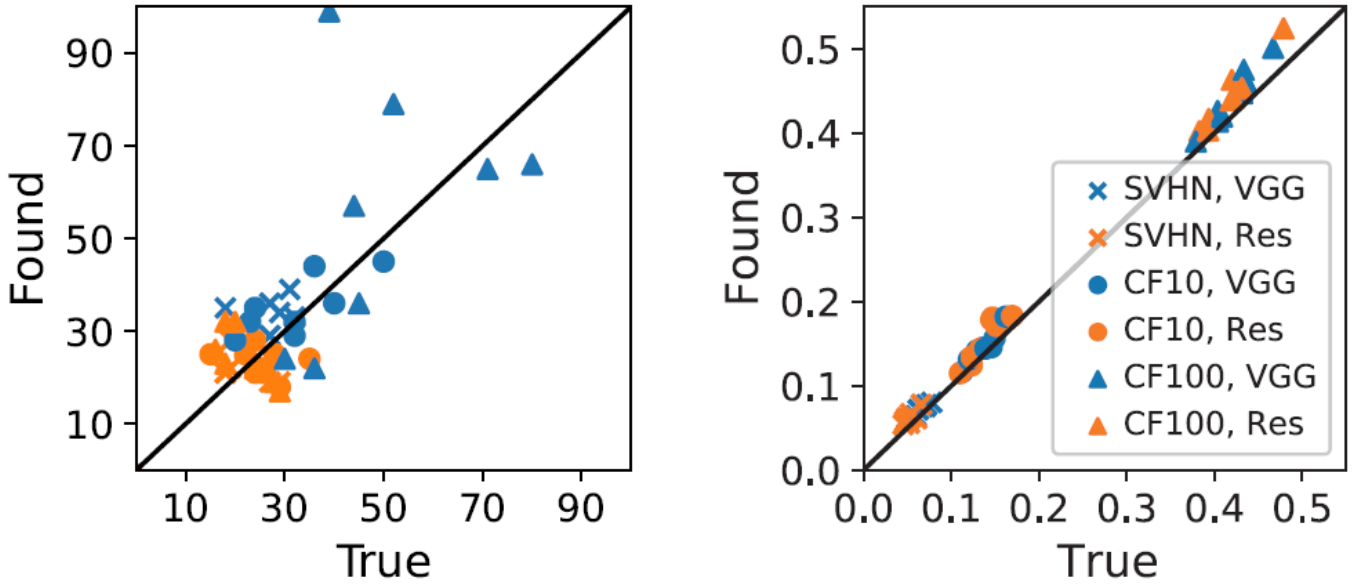
(a) MNIST



(b) FashionMNIST

Even unstable variations of the test accuracy can be reflected by the OV

Application 1: Early Stopping without a Validation Set



(a) Early stopping point

(b) Test error

Figure 4: Early stopping based on test error (True) and the corresponding OV (Found). The shapes represent different datasets, whereas the colors indicate different categories of DNNs ("CF" and "Res" denotes "CIFAR" and "ResNet", respectively).

Application 2: Optimal Network Size Determination

- Train ResNet18 with different network sizes on CIFAR10 for 100 epochs with no label noise.
- For each convolutional layer, set the number of filters k/4 (k=1,2,...,8) times the number of filters in the original model
- The Pearson correlation coefficient between the OV and the test accuracy reached **-0.94**

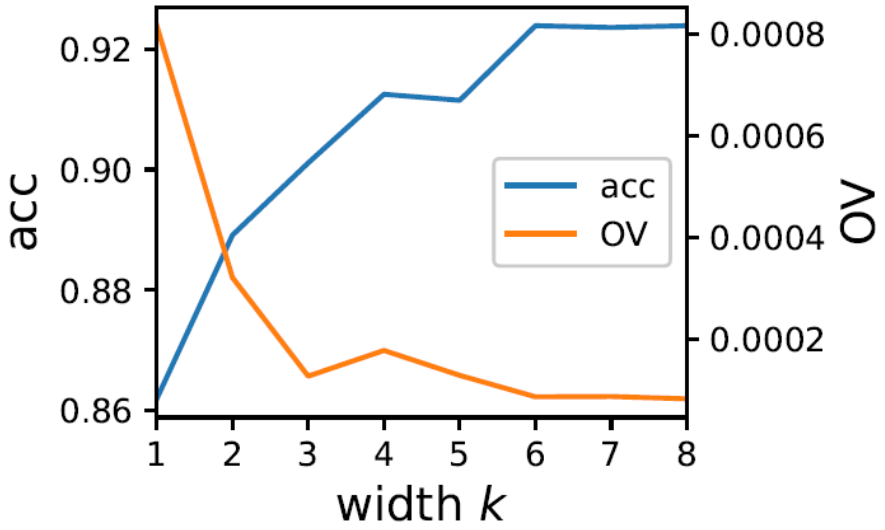


Figure 5: Test accuracy and OV w.r.t. the network size.

Summary

- We show the variance dominates the epoch-wise double descent, and highly correlates with the test error
- We propose optimization variance (OV), which is calculated from the training set only but powerful enough to predict how the test error changes during training
- OV may be used to **perform early stopping without any validation set**, or to **determine the optimal network size**

衷心感谢
CCF-百度松果基金
对本项目的支持！