AI and Deep Learning

3-1 Bias-Variance Analysis

Zhonglei Wang

WISE, SOE, CHOW

XMU

(Version v1)

Contents

Review

1. We have learnt FNNs, and there are two types of parameters:
   - Model parameters: $\{(\boldsymbol{b}^{[l]}, \boldsymbol{W}^{[l]}) : l = 1, \ldots, L\}$
     - They can be estimated by gradient descent algorithms
   - Hyperparameters, which cannot be estimated using training data
2. Our goal changes
   - Do not focus on estimating model parameters by minimizing training errors
   - Concentrate on generalization errors to determine hyperparameters

Hyperparameters

1. α : Learning rate
2. L : Number of layers
3. $\{d^{[l]} : l = 1, \ldots, L-1\}$: Number of neurons per each hidden layer
4. m : Mini-batch size
5. Gradient descent algorithm
6. Number of iterations for the chosen gradient descent algorithm
7. ...

Notations

1. $\boldsymbol{x}$: General notation for a feature vector
2. y : General notation for the observed label
3. $S = \{(\boldsymbol{x}_i, y_i) : i = 1, \ldots, n\}$: Training examples (with existence of randomness)
4. y_t : General notation for the true target given $\boldsymbol{x}$ by a true model$(y_t = E(y \mid \boldsymbol{x}))$
5. $\hat{y}$: Estimation of the true label y_t **based on S** using a working model
6. Both y_t and $\hat{y}_t$ are functions of $\boldsymbol{x}$

From a least-squared-error perspective

1. Given a feature $\boldsymbol{x}$

$$
\begin{aligned}
E_S\{(\hat{y}-y_t)^2 \mid \boldsymbol{x}\} &= E_S[\{\hat{y}-E_S(\hat{y}\mid \boldsymbol{x})+E_S(\hat{y}\mid \boldsymbol{x})-y_t\}^2 \mid \boldsymbol{x}] \\
&= E_S[\{E_S(\hat{y}\mid \boldsymbol{x})-y_t\}^2+2\cdot\{E_S(\hat{y}\mid \boldsymbol{x})-y_t\}\cdot\{\hat{y}-E_S(\hat{y}\mid \boldsymbol{x})\}+\{\hat{y}-E_S(\hat{y}\mid \boldsymbol{x})\}^2 \mid \boldsymbol{x}] \\
&= E_S[\{E_S(\hat{y}\mid \boldsymbol{x})-y_t\}^2 \mid \boldsymbol{x}]+2\cdot\{E_S(\hat{y}\mid \boldsymbol{x})-y_t\}\cdot E_S[\{\hat{y}-E_S(\hat{y}\mid \boldsymbol{x})\}\mid \boldsymbol{x}] \\
&\quad + E_S[\{\hat{y}-E_S(\hat{y}\mid \boldsymbol{x})\}^2 \mid \boldsymbol{x}] \\
&= \{E_S(\hat{y}\mid \boldsymbol{x})-y_t\}^2+E_S[\{\hat{y}-E_S(\hat{y}\mid \boldsymbol{x})\}^2 \mid \boldsymbol{x}] \\
&= (\text{Bias})^2+\text{Variance}
\end{aligned}
$$

- $E_S(\cdot)$: Expectation with respect to the randomness associated with S
- $E_S(\hat{y}\mid \boldsymbol{x})-y_t$: Nonrandom, the error between y_t and a pre-specified model, namely the bias
- $\hat{y}-E_S(\hat{y}\mid \boldsymbol{x})$: Random, measures the model's variability with respect to S

Bias and variance

1. Bias

$$\text{Bias}(\hat{y}) = E_S(\hat{y} \mid \boldsymbol{x}) - y_t$$

- $E_S(\cdot)$: Expectation with respect to the randomness associated with S

2. Variance

$$\text{Variance}(\hat{y}) = E_S[\{\hat{y} - E_S(\hat{y} \mid \boldsymbol{x})\}^2 \mid \boldsymbol{x}]$$

3. Bias and variance are defined for a GIVEN feature $\boldsymbol{x}$

Examples--Mean estimation

1. Consider the following setup

$$y \mid \boldsymbol{x} = \mu + \epsilon$$

- $E(\epsilon \mid \boldsymbol{x}) = 0, \quad \text{Variance}(\epsilon \mid \boldsymbol{x}) = \sigma^2$
- True regression is a constant function with respect to the feature $\boldsymbol{x}$
- True label $y_t = E(y \mid \boldsymbol{x}) = \mu$

2. For a new feature $\boldsymbol{x}$, the label is estimated

$$\hat{y} = \hat{\mu}$$

- $\hat{\mu} = n^{-1} \sum_{i=1}^{n} y_i$
- The (working) model is $f(\boldsymbol{x}) = c$ for all $\boldsymbol{x}$, where c is the model parameter (constant)

Examples--Mean estimation

1. Bias

$$\begin{aligned}\text{Bias}(\hat{y}) &= E_S(\hat{y} \mid \boldsymbol{x}) - y_t \\ &= E_S(\hat{\mu}) - \mu = 0\end{aligned}$$

2. Variance

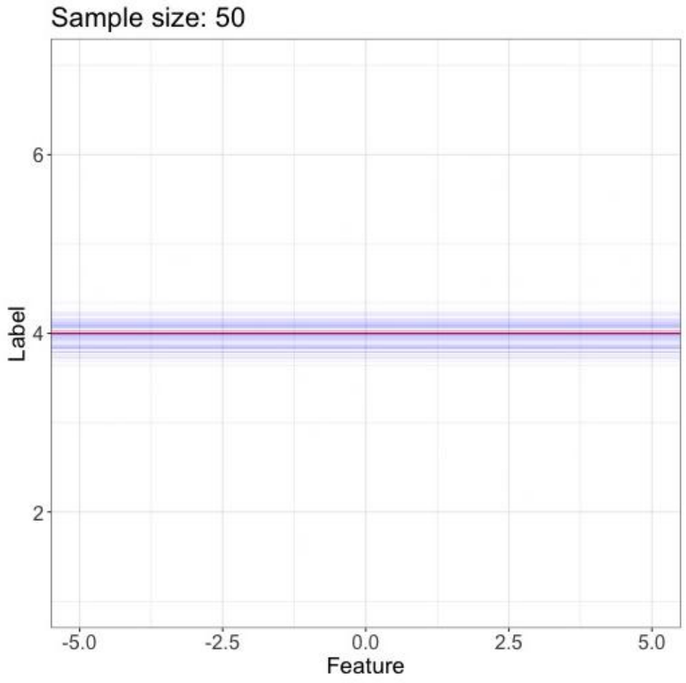
$$\begin{aligned}\text{Variance}(\hat{y}) &= E_S[\{\hat{y} - E_S(\hat{y})\}^2 \mid \boldsymbol{x}] \\ &= \text{Variance}(\hat{\mu}) \\ &= n^{-1}\sigma^2\end{aligned}$$

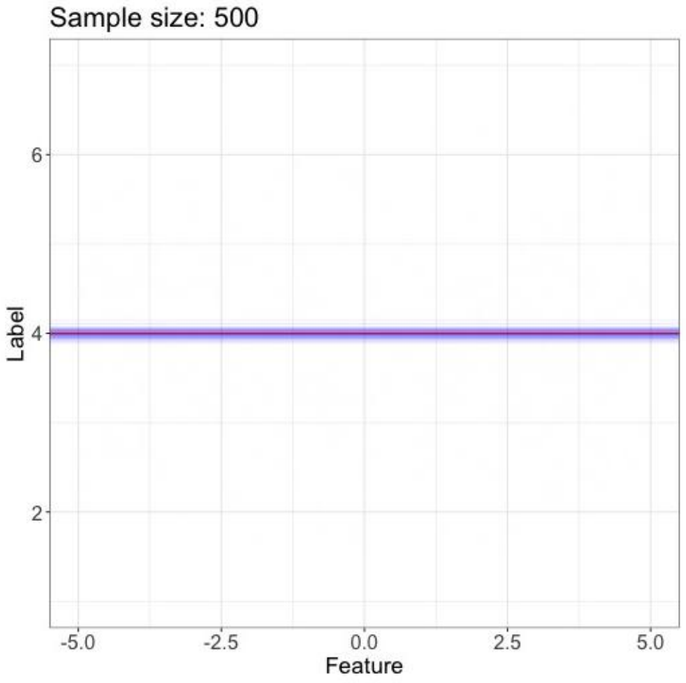
3. Those properties are what we have learnt for the mean estimator

Examples--Mean estimation

1. A toy example

- $x_i \sim \text{Uniform}[-5, 5], \quad y_i \mid x_i = 6 + \epsilon_i, \quad \epsilon_i \sim N(0, 1^2) \quad (i = 1, \ldots, n)$





Examples--Linear regression

1. Consider the following setup

$$y \mid \boldsymbol{x} = b_0 + \boldsymbol{x}^{\mathrm{T}} \cdot \boldsymbol{w}_0 + \epsilon$$

- True regression is a linear function of the feature $\boldsymbol{x}$ with parameters $b_0, \boldsymbol{w}_0$
- True label $y_t = E(y \mid \boldsymbol{x}) = b_0 + \boldsymbol{x}^{\mathrm{T}} \boldsymbol{w}_0$
- $E(\epsilon \mid \boldsymbol{x}) = 0, \quad \text{Variance}(\epsilon \mid \boldsymbol{x}) = \sigma^2$

Examples--Linear regression

1. For a new feature $\boldsymbol{x}$, the estimated label is

$$\hat{y} = \hat{b} + \boldsymbol{x}^{\mathrm{T}} \cdot \hat{\boldsymbol{w}}$$

- The (working) model is $f(\boldsymbol{x}; \boldsymbol{\theta}) = b + \boldsymbol{x}^{\mathrm{T}} \boldsymbol{w}$, with model parameter $\boldsymbol{\theta} = (b, \boldsymbol{w}^{\mathrm{T}})^{\mathrm{T}}$
- $\hat{b}, \hat{\boldsymbol{w}}$: Estimated by minimizing

$$n^{-1} \sum_{i=1}^{n} (y_i - b - \boldsymbol{x}^{\mathrm{T}} \cdot \boldsymbol{w})^2$$

- Check Chapter 1 for the solution

Examples--Linear regression

1. We can show

$$E_S(\hat{b}) = b_0 \quad E_S(\hat{\boldsymbol{w}}) = \boldsymbol{w}_0$$

- That is, the estimated model parameters are unbiased.

2. Bias

$$\begin{aligned}\text{Bias}(\hat{y}) &= E_S(\hat{y} \mid \boldsymbol{x}) - y_t \\ &= E_S(\hat{b} + \boldsymbol{x}^{\mathrm{T}} \cdot \hat{\boldsymbol{w}} \mid \boldsymbol{x}) - (b_0 + \boldsymbol{x}^{\mathrm{T}} \cdot \boldsymbol{w}_0) = 0\end{aligned}$$

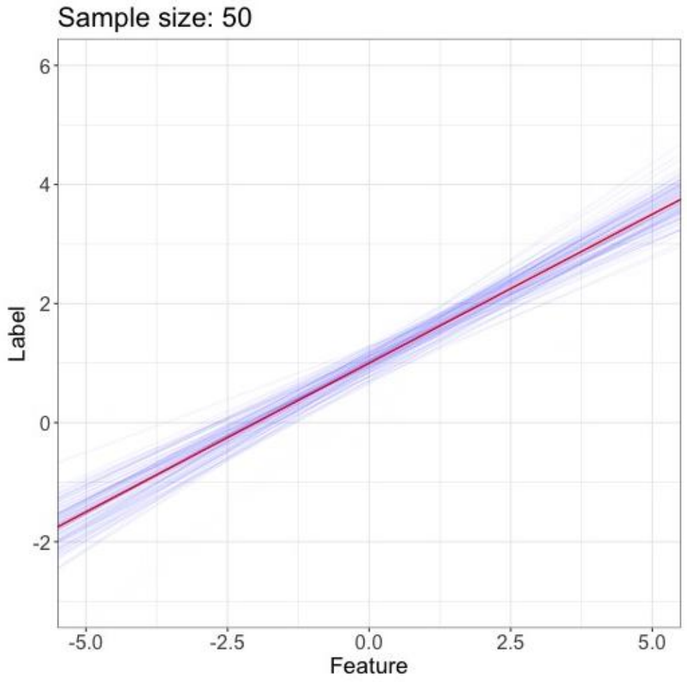
3. Variance

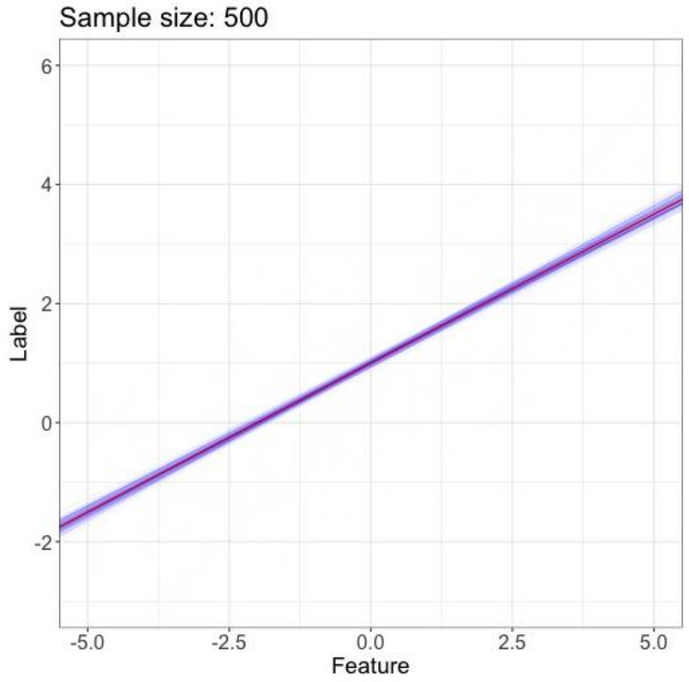
$$\begin{aligned}\text{Variance}(\hat{y}) &= E_S[\{\hat{y} - E_S(\hat{y})\}^2 \mid \boldsymbol{x}] \\ &= \text{Variance}(\hat{b} + \boldsymbol{x}^{\mathrm{T}} \cdot \hat{\boldsymbol{w}} \mid \boldsymbol{x}) = (\text{Check your textbook})\end{aligned}$$

Examples--Linear regression

1. A toy example

- $x_i \sim \text{Uniform}[-5, 5], \quad y_i \mid x_i = 1 + 0.5x_i + \epsilon_i, \quad \epsilon_i \sim N(0, 1^2) \quad (i = 1, \ldots, n)$





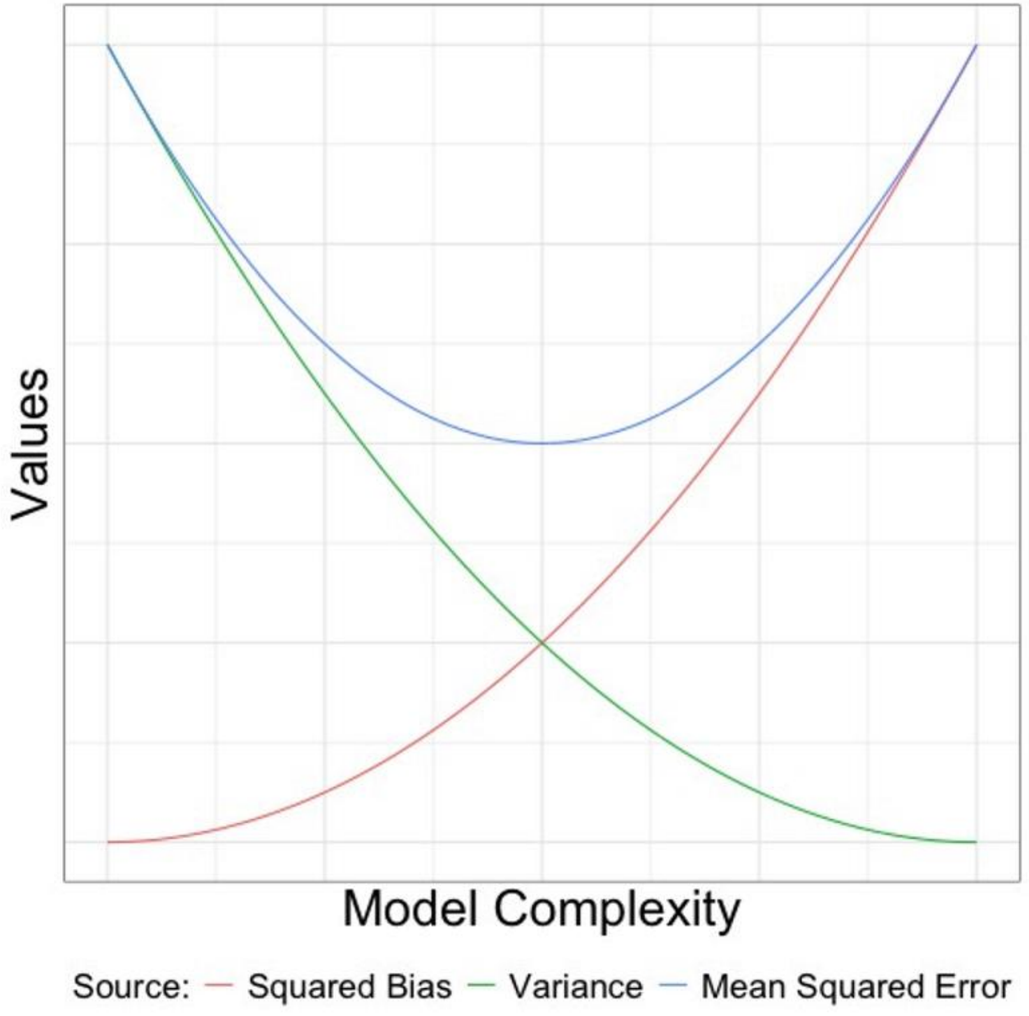
Examples--Ridge regression

1. We still consider the setup for linear regression
2. Model parameters are estimated by minimizing

$$\sum_{i=1}^{n}(y_i - b - \boldsymbol{x}_i^{\mathrm{T}} \cdot \boldsymbol{w})^2 + \lambda \sum_{j=1}^{d} w_j^2$$

- $\boldsymbol{w} = (w_1, \ldots, w_d)^{\mathrm{T}}$
- Hyperparameter λ to control the complexity of the model
- Estimated label is no longer unbiased, and check textbook for more discussion

Classical bias-variance tradeoff



Tune the hyperparameters for classical models

1. Cross validation is used for tradition statistical models
2. It is not feasible for tuning hyperparameters in deep learning models
3. For deep learning models, we use a validation set to tune hyperparameters
   - Training dataset: Train a deep learning model
   - Validation dataset: Evaluate the performance with different hyperparameters
     - ▷ Different sets of hyperparameters correspond to different models
     - ▷ Choosing a good set of hyperparameters is equivalent to finding a good model
   - Test dataset (optional): Test the performance of the CHOSEN model in reality

Tune hyperparameters

1. Criterion:
   - Reduce bias first
     - Increase training dataset (expensive)
     - Consider more complex models
   - If the bias is controlled, reduce the variance
     - Increase training dataset (expensive)
     - Regularization

Double descent

1. Traditionally,
   - Simpler models corresponds to large bias and small variance
   - More sophisticated models corresponds to small bias and large variance
2. By classical theories, as model complexity increases,
   - Bias decreases
   - Variance increases
   - Thus, "larger models are worse!"

Double descent

1. Deep learning models have fundamentally challenged the classical bias-variance tradeoff theory
   - "As we increase model size, performance first gets worse and then gets better"
   - 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
   - Check the paper by Belkin et al. (2019) Nakkiran et al. (2019) for details

Double descent

1. The following is Figure 1 of Nakkiran et al. (2019)

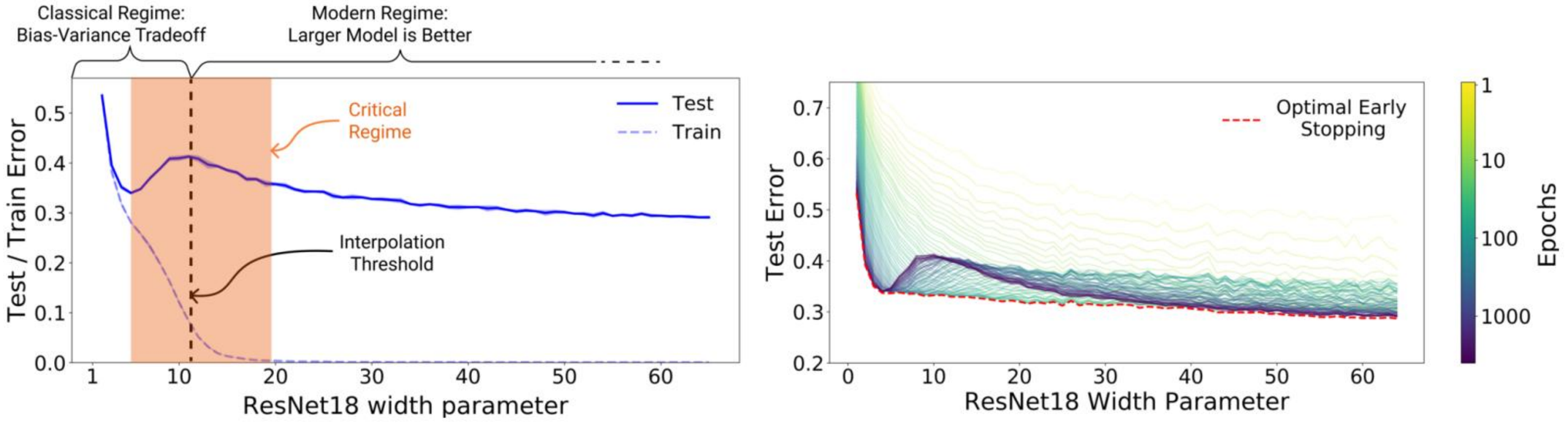


Figure 1: **Left:** Train and test error as a function of model size, for ResNet18s of varying width on CIFAR-10 with 15% label noise. **Right:** Test error, shown for varying train epochs. All models trained using Adam for 4K epochs. The largest model (width 64) corresponds to standard ResNet18.

Double descent

1. Three regions for generalization errors:
   - Traditional region: Number of parameters is much smaller than the training examples
   - Critical region: Number of parameters is comparable with that of training examples
   - Modern region: Number of parameters is much larger with that of training examples
2. More training examples may undermine the performance of deep learning models
   - For an over-parameterized model, more training examples may drag the model from modern region to critical region
   - We should increase the complexity of the model accordingly

Double descent

1. The following is Figure 2 of Nakkiran et al. (2019)

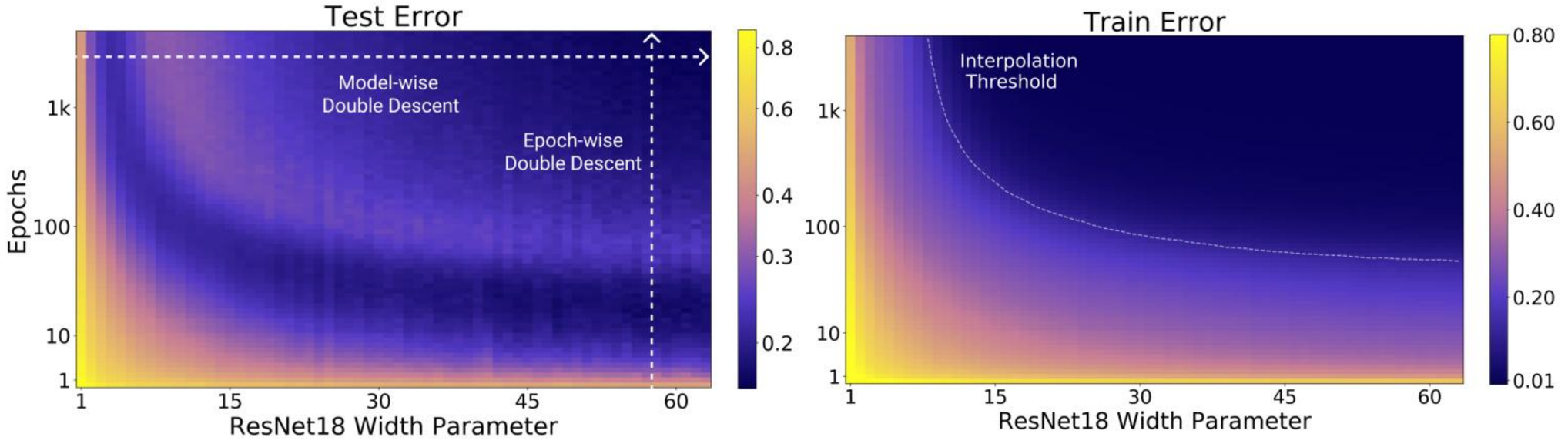


Figure 2: **Left:** Test error as a function of model size and train epochs. The horizontal line corresponds to model-wise double descent–varying model size while training for as long as possible. The vertical line corresponds to epoch-wise double descent, with test error undergoing double-descent as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

Summary

1. Hyperparameters are important for the performance of a model
2. Bias-variance analysis can be used to select hyperparameters
3. For modern deep learning, bias-variance analysis is not the only truth
4. What matters is the regime it lies in and the training dynamics