3-1 Bias-Variance Analysis

Zhonglei Wang

WISE and SOE, XMU, 2025

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

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. $\cdots$

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
4. y_t : general notation for the true target given $\boldsymbol{x}(y_t = E(y \mid \boldsymbol{x}))$
5. $\hat{y}$: estimation of the true label y_t based on S using a certain model

Bias and variance

1. Bias

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

- $E_S(\cdot)$: expectation with respect to the randomness existed in generating S
- Bias and variance are defined for a GIVEN feature $\boldsymbol{x}$

2. Variance

$$\text{Variance}(\hat{y}) = E_S\{\hat{y} - E_S(\hat{y})\}^2$$

Examples -- mean estimation

1. Consider the following setup

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

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

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

2. Variance

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

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

Examples -- linear regression

1. Consider the following setup

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

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

Examples -- linear regression

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

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

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

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

- 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

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

3. Variance

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

Examples -- ridge regression

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

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

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

Double descent

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

Double descent

1. Surprisingly, for deep learning models, we have the amazing double descent phenomen
   - "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 Nakkiran et al. (2019) for details

Double descent

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

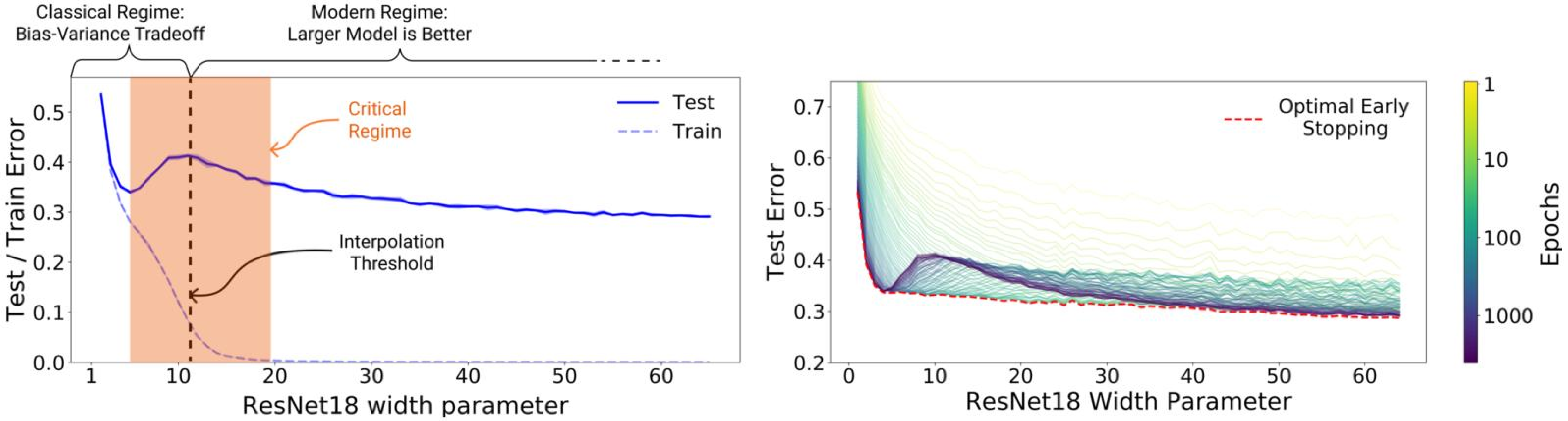

Figure 1: **Left:** Train and test error as a function of model size, for ResNet18s of varying width on CIFAR-10 with 15% label noise. **Right:** Test error, shown for varying train epochs. All models trained using Adam for 4K epochs. The largest model (width 64) corresponds to standard ResNet18.

Double descent

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

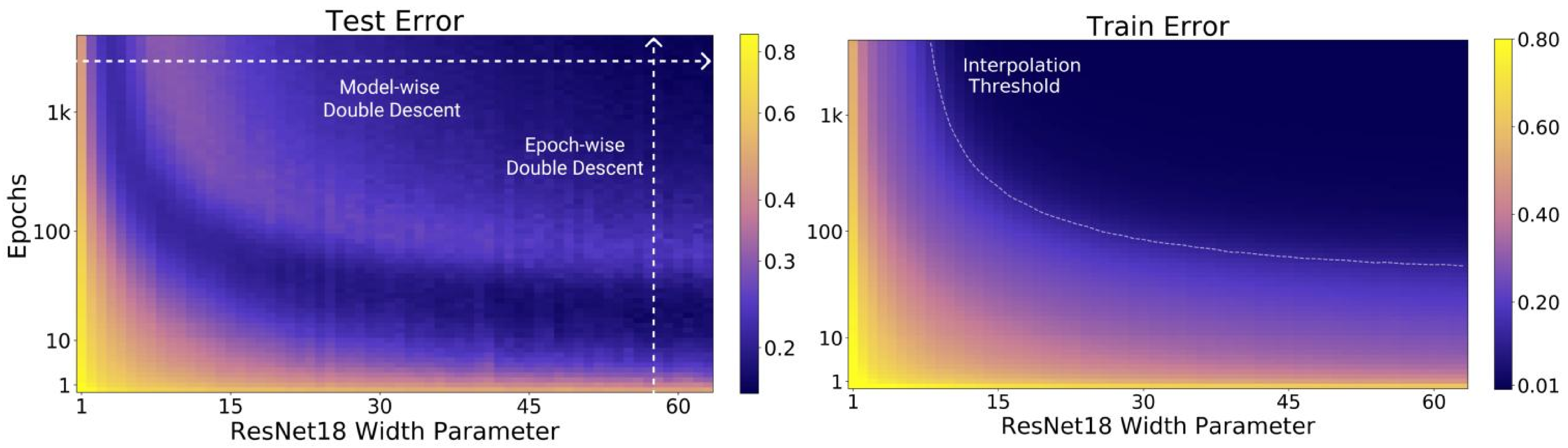

Figure 2: **Left:** Test error as a function of model size and train epochs. The horizontal line corresponds to model-wise double descent–varying model size while training for as long as possible. The vertical line corresponds to epoch-wise double descent, with test error undergoing double-descent as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

Tune the hyperparameters

1. Cross validation is used for tradition statistical models
2. It is not feasible for deep learning models
3. For deep learning models, we use a validation set to tune hyperparameters
   - Training dataset: to train a deep learning model
   - Validation dataset: evaluate the performance of models 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 real application

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