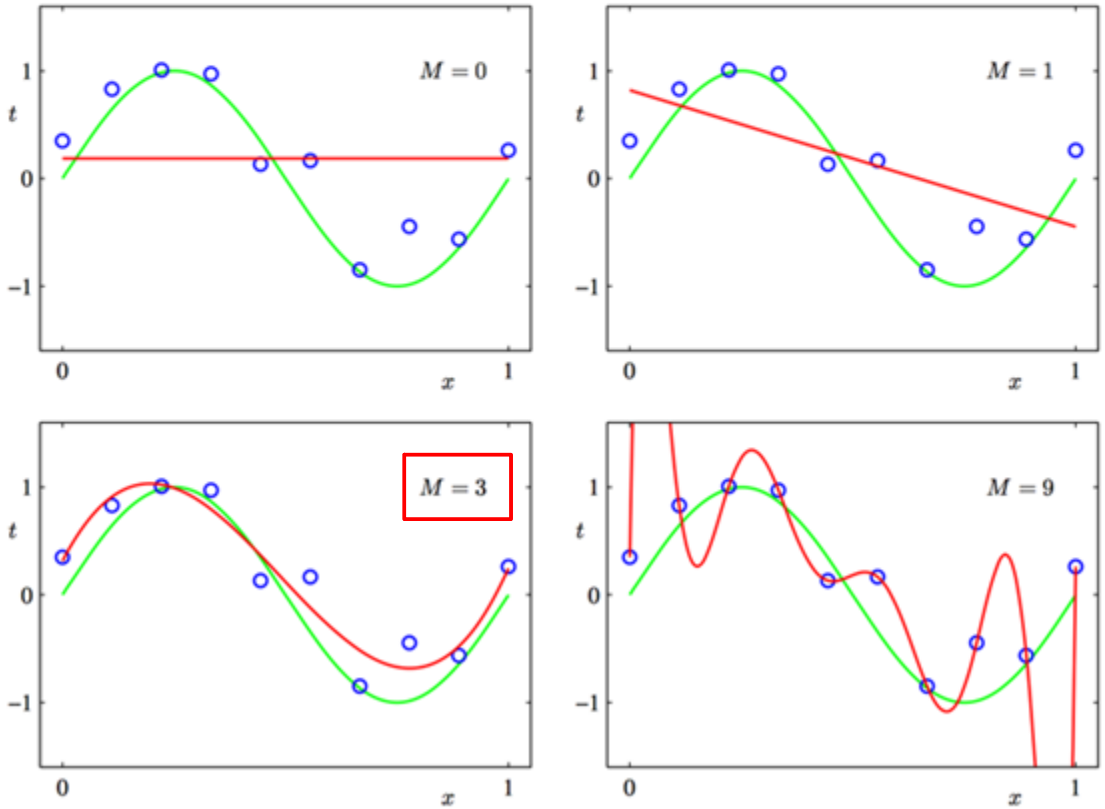
Regularization

$y = \sum_{0}^{M} \beta_i x^i$ $\begin{cases} M=0, & y=\beta_0 \\ M=1, & y=\beta_0+\beta_1 x \\ \vdots \end{cases}$

Overfitting and Underfitting

- Machine learning polynomial function
  - What is a good model?

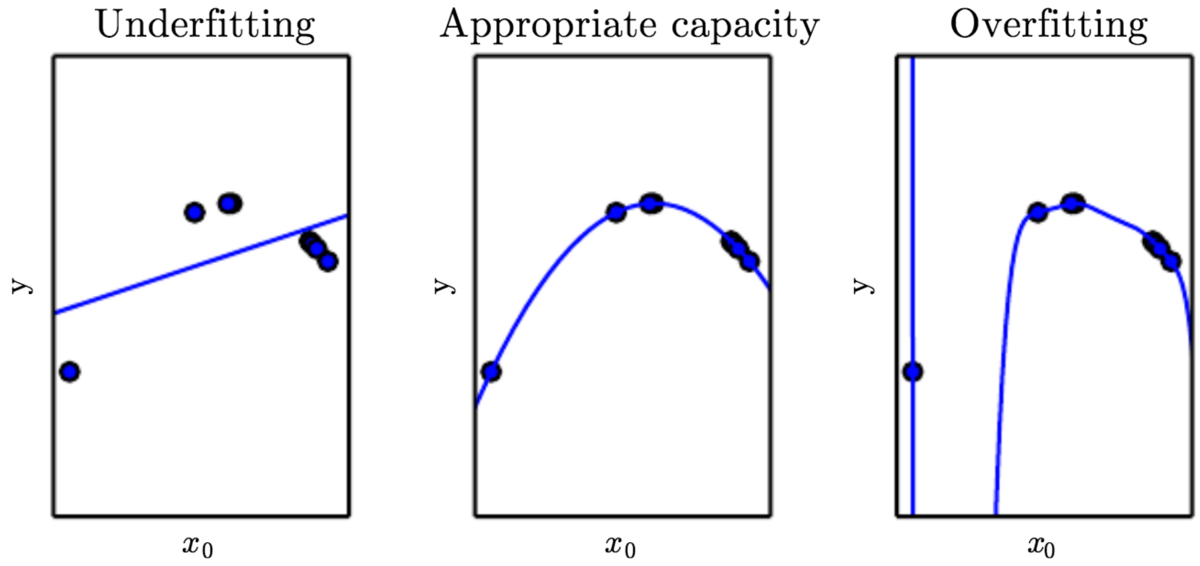


Overfitting and Underfitting

- Our algorithm must perform well on new, previously unseen inputs.
  - The ability to perform well on them is called generalization.
- Typically, we have access to a training set when training a machine learning model.
  - Training, and then computing some error measure on the training set → reduce the training error
  - We want the generalization error (also called the test error) to be low as well.
  - The generalization error = the expected value of the error on a new input
  - We estimate the generalization error by measuring the performance on a test set collected separately from the training set. test set ∩ training set = ∅

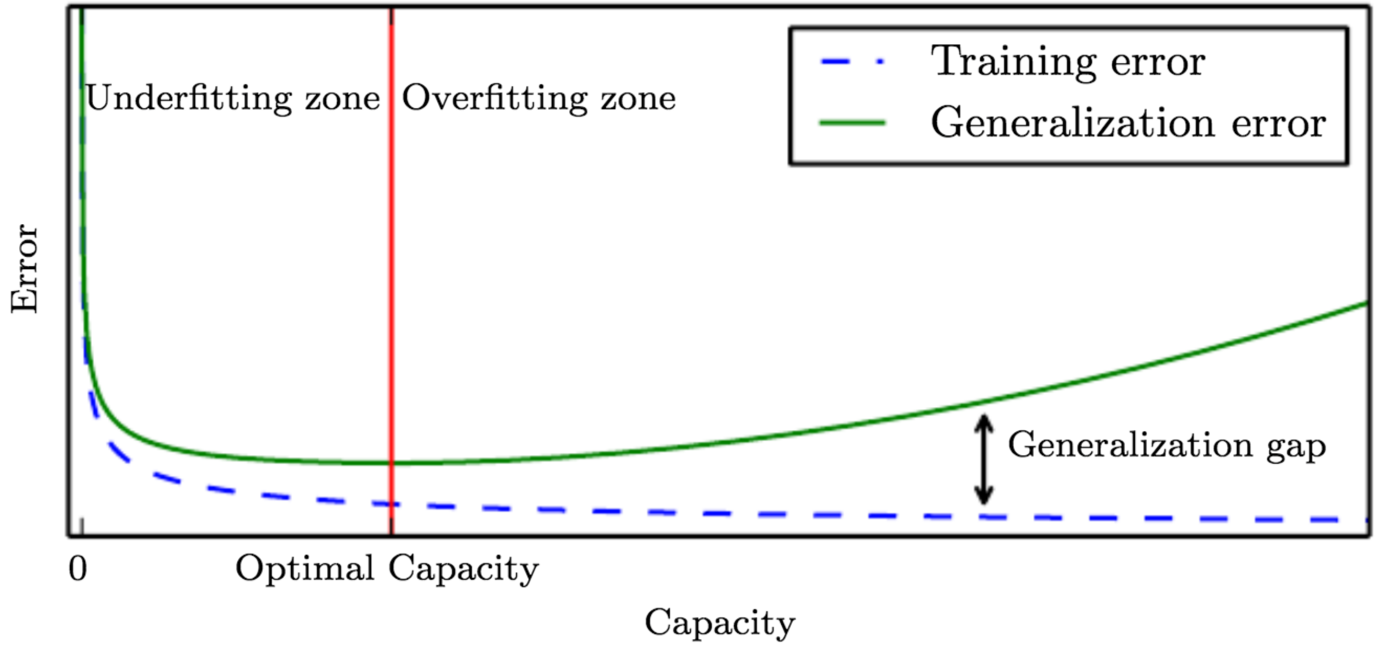
Overfitting and Underfitting

- The factors determining how well a ML algorithm will perform are its ability to
  - make the training error small → underfitting (training error 감소 실패)
  - make the gap between training and test error small → overfitting(training error와 test error 사이 격차 큼)



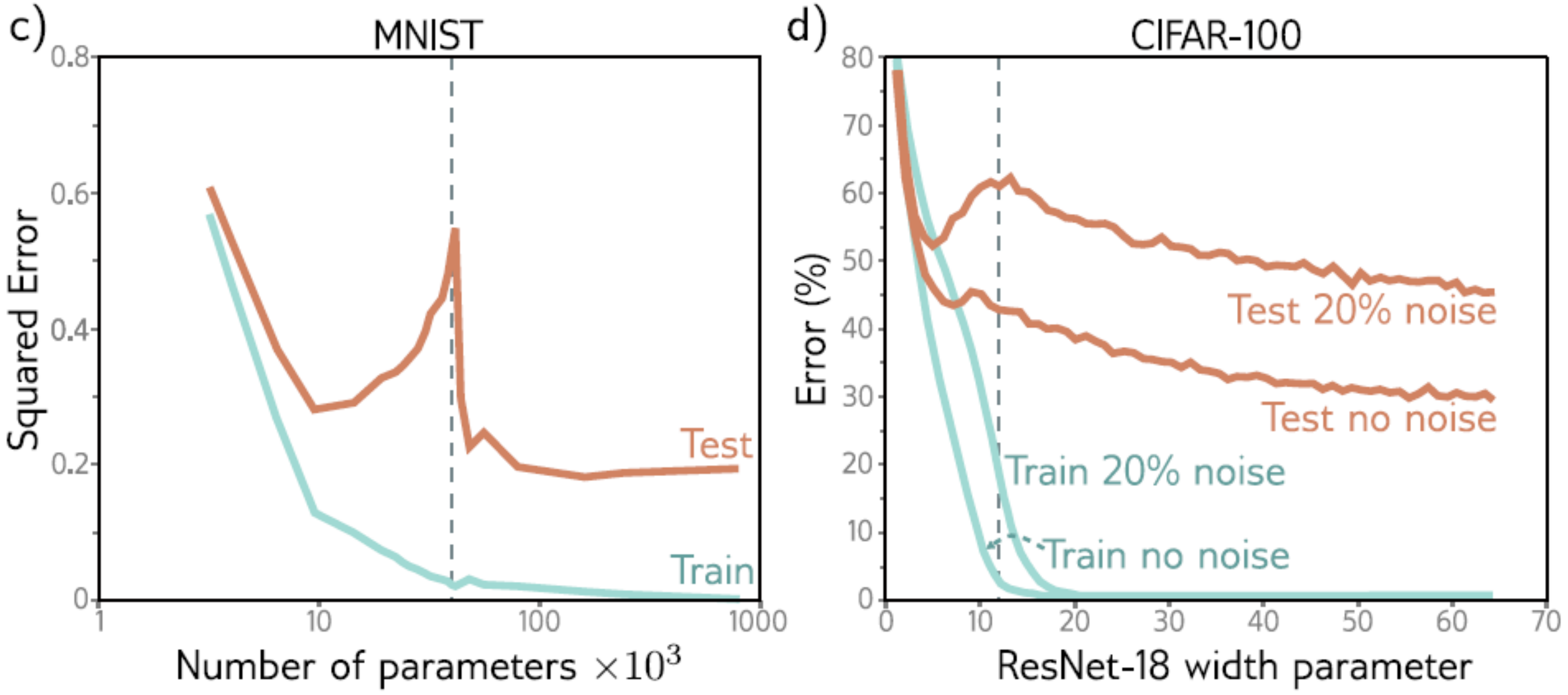
Sangheum Hwang

Overfitting and Underfitting



This is conventional wisdom… → The *double descent* phenomenon

Double descent phenomenon



Unexpected behavior. Why this happens?

Refer to Section 8.4 in “Understanding deep learning”

Overfitting and Underfitting

- How to avoid overfitting?
- Regularization
  - Any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error
  - Example
    - Adding a penalty term called a regularizer $\Omega(\boldsymbol{\beta})$ to the cost function

$$J(\boldsymbol{\beta}) = \mathrm{MSE}_{\mathrm{train}} + \lambda\Omega(\boldsymbol{\beta})$$

Overfitting and Underfitting

- Regularization
  - Example: weight decay $\Omega(\boldsymbol{\beta}) = \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\beta}$, also called L2 regularization

$$J\boxed{(\boldsymbol{\beta})} = \mathrm{MSE}_{\mathrm{train}} + \boxed{\lambda}\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\beta}$$

For linear regression, it is called ridge regression.

parameters of model

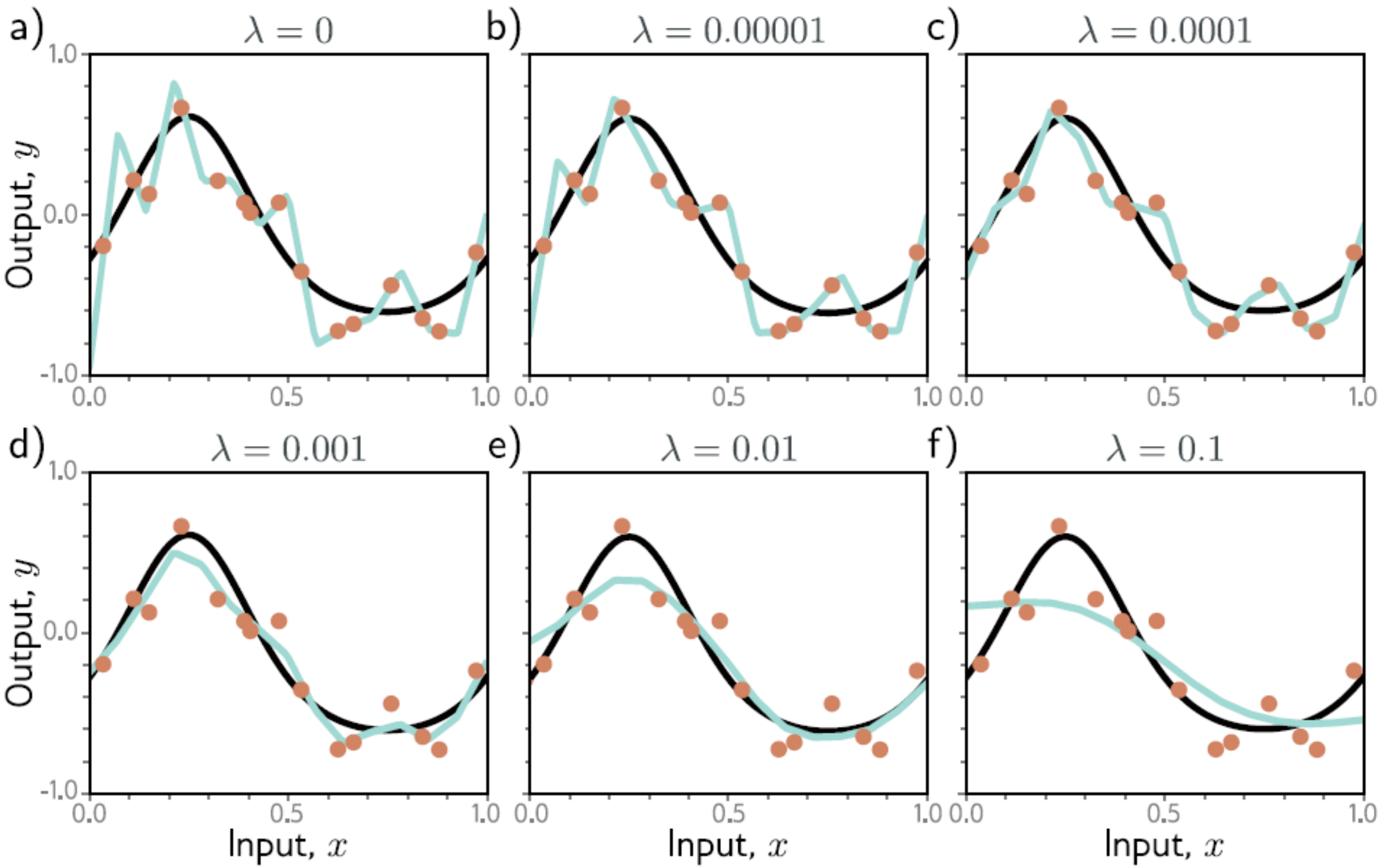
hyperparameter, regularization 강도 조절.

  - Example: L1 regularization $\Omega(\boldsymbol{\beta}) = ||\boldsymbol{\beta}||_1$

$$J(\boldsymbol{\beta}) = \mathrm{MSE}_{\mathrm{train}} + \lambda||\boldsymbol{\beta}||_1$$

For linear regression, it is called lasso regression.

L2 regularization example



Hyperparameters and validation sets

model parameter : β
hyperparameter : λ

- Most machine learning algorithms have hyperparameters that control the algorithm's behavior.
  - Linear regression has no hyperparameter.
- It is not appropriate to choose hyperparameters based on the training set.
  - This will result in overfitting.
- We need a validation set, which consists of data points that were not used for training.
  - So, the hyperparameters showing the lowest validation error will be chosen.

Training, validation, and test

- Training set: to learn the parameters of the model
- Validation set: to choose the hyperparameters of the model
- Test set: for final evaluation of the generalization error of the model
  - How well will our model perform with new data that were not observed during training and validation?