Something before Real Deep Learning - Overfitting

Yizhen Lao

Framework of ML

Training data: $\{(\boldsymbol{x}^1, \hat{y}^1), (\boldsymbol{x}^2, \hat{y}^2), \ldots, (\boldsymbol{x}^N, \hat{y}^N)\}$

Testing data: $\{\boldsymbol{x}^{N+1}, \boldsymbol{x}^{N+2}, \ldots, \boldsymbol{x}^{N+M}\}$

Speech Recognition

$\boldsymbol{x}$:  $\hat{y}$: phoneme

Image Recognition

$\boldsymbol{x}$:  $\hat{y}$: soup

Speaker Recognition

$\boldsymbol{x}$:  $\hat{y}$: John (speaker)

Machine Translation

$\boldsymbol{x}$: 痛みを知れ

$\hat{y}$: 了解痛苦吧

Framework of ML

Training data: $\{(\boldsymbol{x}^1, \hat{y}^1), (\boldsymbol{x}^2, \hat{y}^2), \ldots, (\boldsymbol{x}^N, \hat{y}^N)\}$

Training:

Step 1: function with unknown → Step 2: define loss from training data → Step 3: optimization

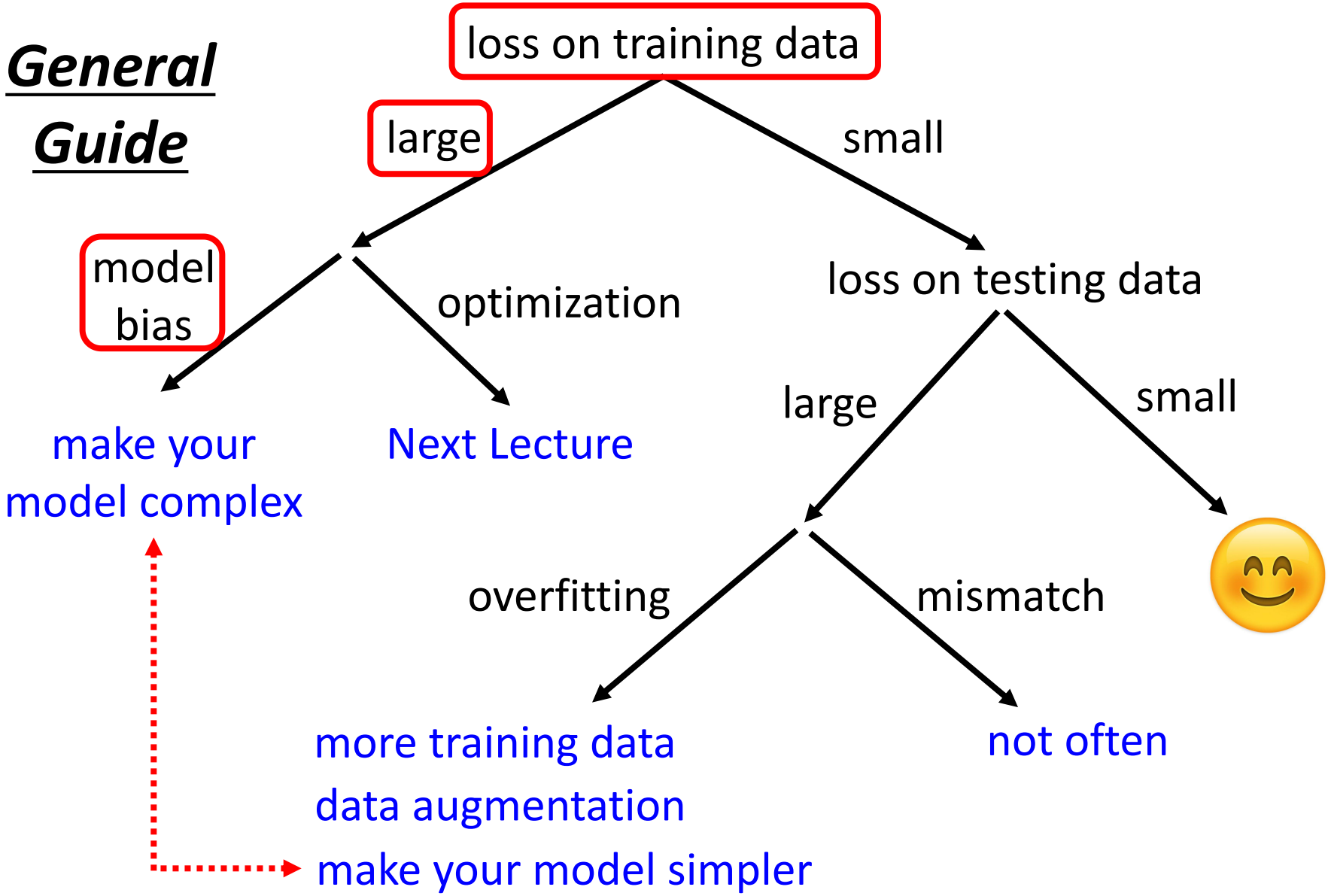
$y = f_{\boldsymbol{\theta}}(\boldsymbol{x})$ $\quad$ $L(\boldsymbol{\theta})$ $\quad$ $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} L$

Testing data: $\{\boldsymbol{x}^{N+1}, \boldsymbol{x}^{N+2}, \ldots, \boldsymbol{x}^{N+M}\}$

Use $y = f_{\boldsymbol{\theta}^*}(\boldsymbol{x})$ to label the testing data

$\{y^{N+1}, y^{N+2}, \ldots, y^{N+M}\}$ → Upload to Kaggle

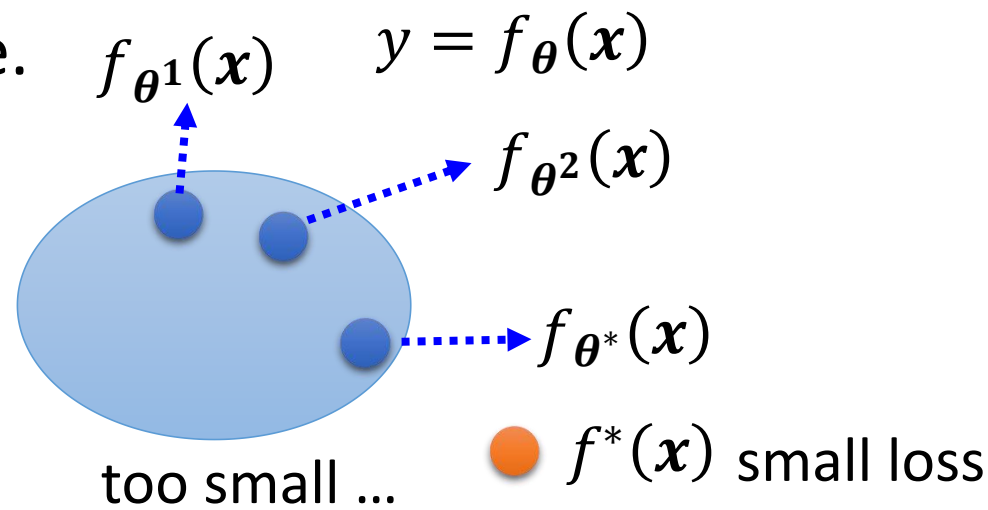
General Guide



Split your training data into training set and validation set for model selection

Model Bias

- The model is too simple.



find a needle in a haystack …

… but there is no needle

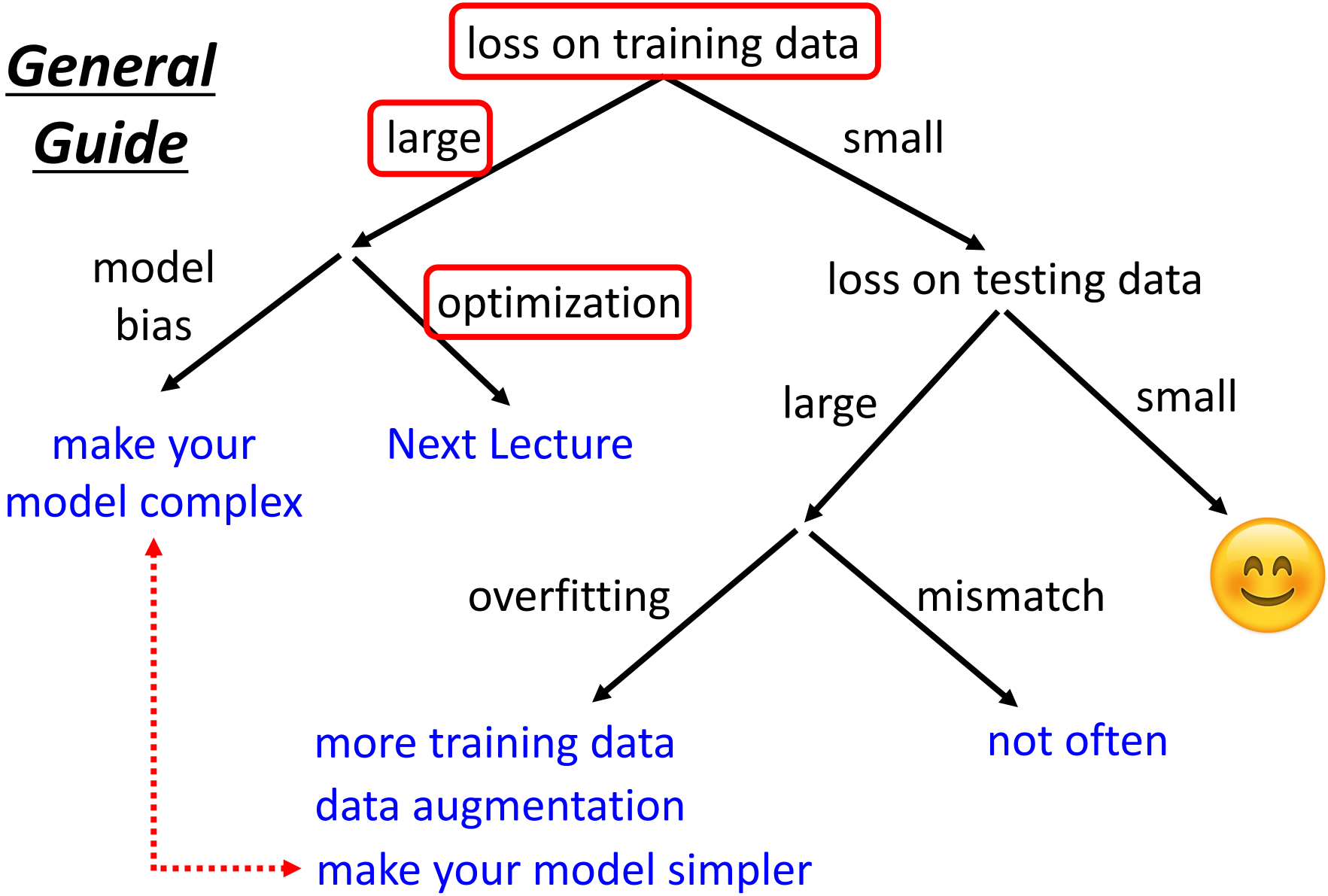
- Solution: redesign your model to make it more flexible

$$y = b + wx_1 \xrightarrow{\text{More features}} y = b + \sum_{j=1}^{56} w_j x_j$$

Deep Learning
(more neurons, layers)

$$y = b + \sum_i c_i \, sigmoid\left(b_i + \sum_j w_{ij} x_j\right)$$

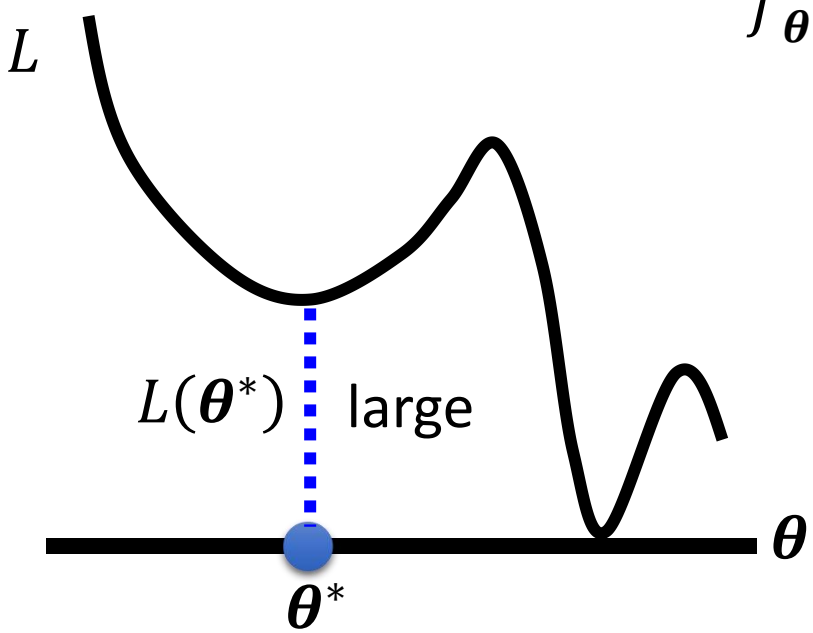
General Guide

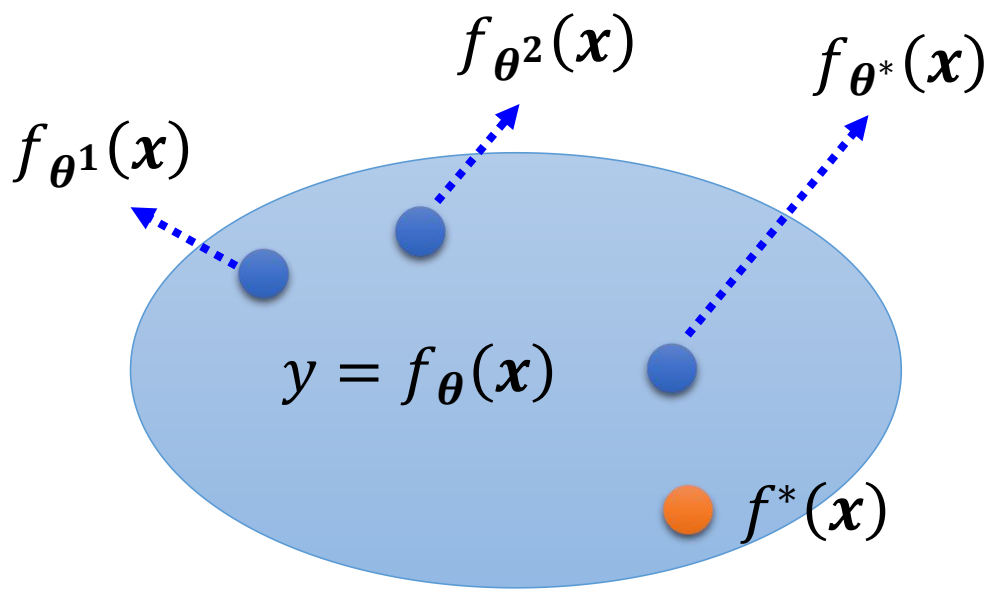


Split your training data into training set and validation set for model selection

Optimization Issue

- Large loss not always imply model bias. There is another possibility …





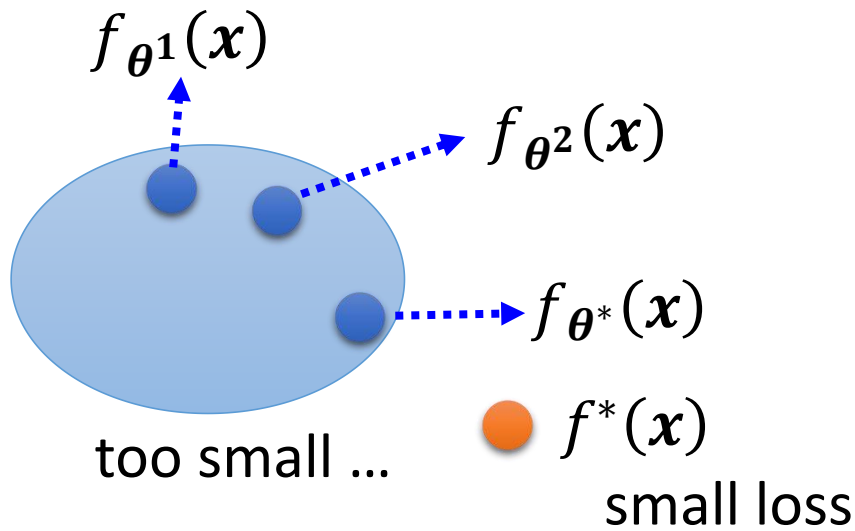
A needle is in a haystack …

… Just cannot find it.

Model Bias

find a needle in a haystack …

… but there is no needle

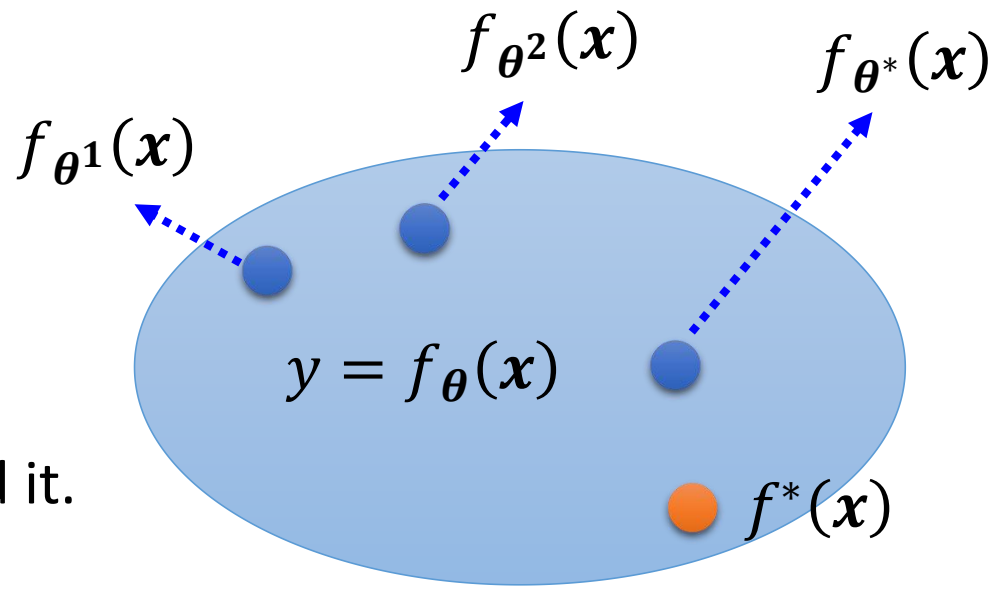


Which one???

Optimization Issue

A needle is in a haystack …

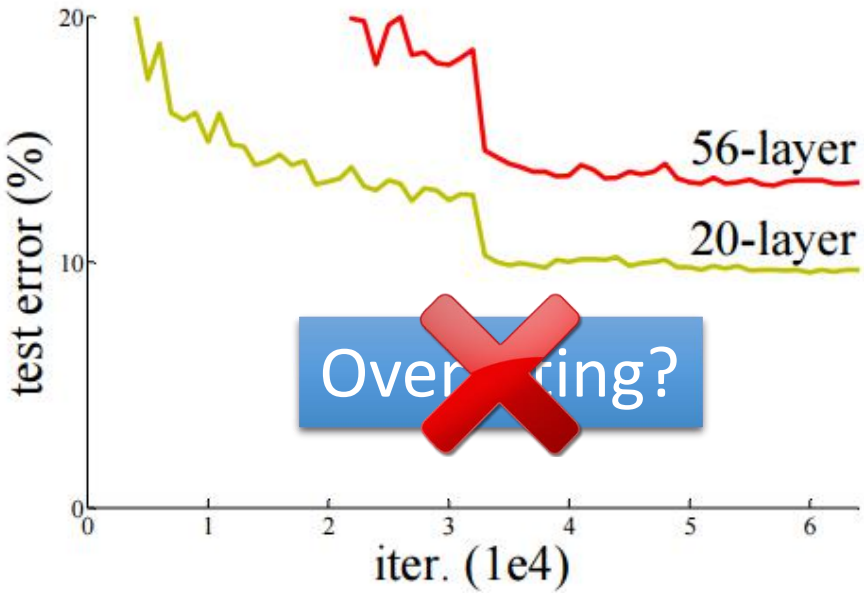
… Just cannot find it.



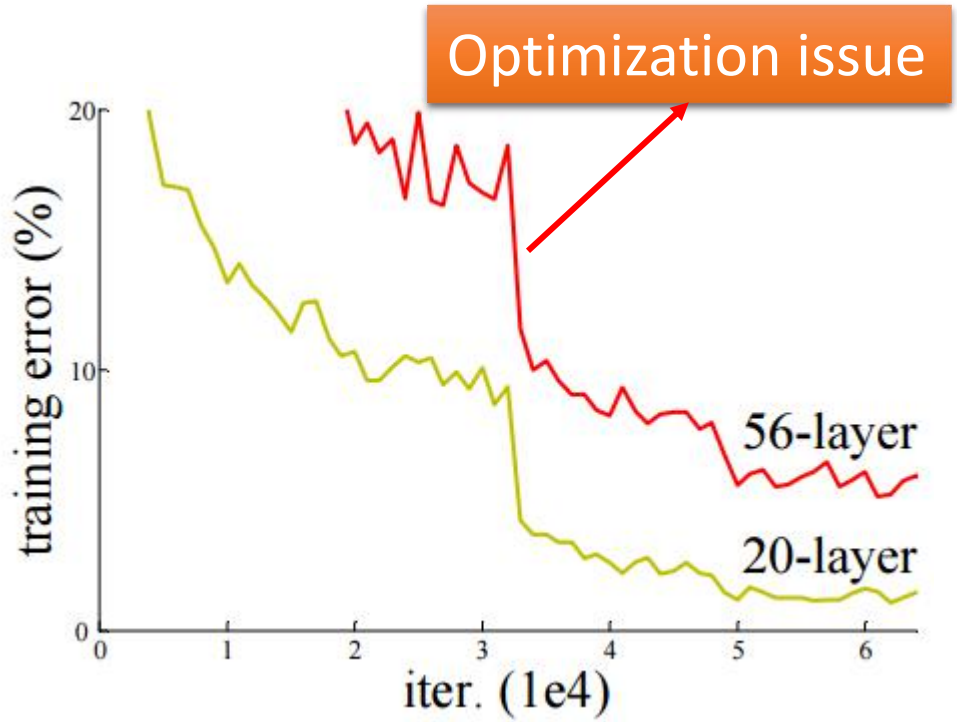
Ref:
http://arxiv.org/abs/1512.03385

Model Bias v.s. Optimization Issue

- Gaining the insights from comparison



Testing Data



Training Data

Ref:
http://arxiv.org/abs/1512.03385

Optimization Issue

- Gaining the insights from comparison
- Start from shallower networks (or other models), which are easier to optimize.
- If deeper networks do not obtain smaller loss on **training data**, then there is optimization issue.

	1 layer	2 layer	3 layer	4 layer	5 layer
2017 – 2020	0.28k	0.18k	0.14k	0.10k	0.34k

- Solution: More powerful optimization technology (next lecture)

General Guide

loss on training data

large

small

model bias

optimization

loss on testing data

make your model complex

Next Lecture

large

small

overfitting

mismatch

more training data

data augmentation

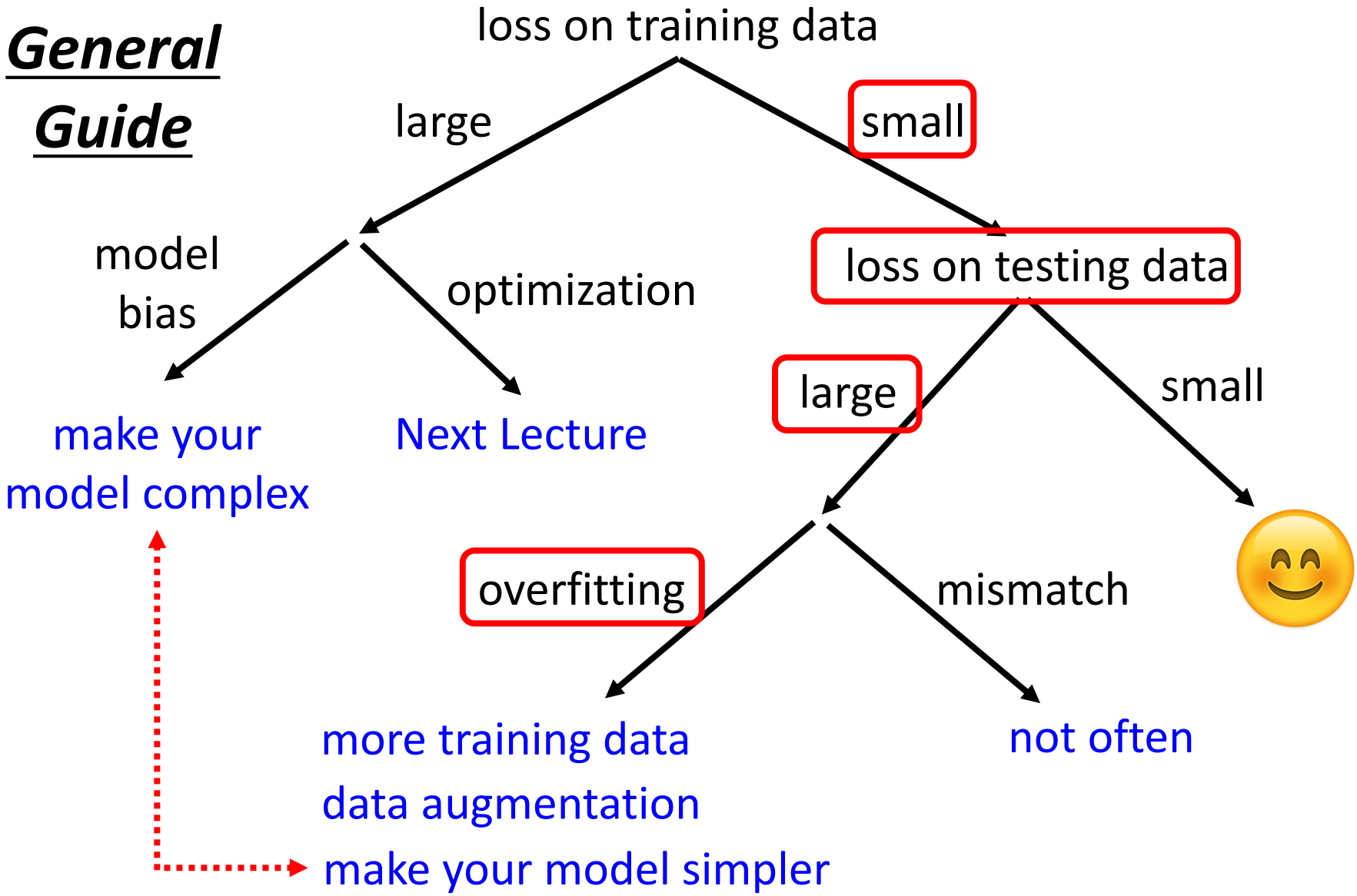
not often

make your model simpler

trade-off

Split your training data into training set and validation set for model selection

General Guide



Split your training data into training set and validation set for model selection

Overfitting

- Small loss on training data, large loss on testing data. Why?

<u>An extreme example</u>

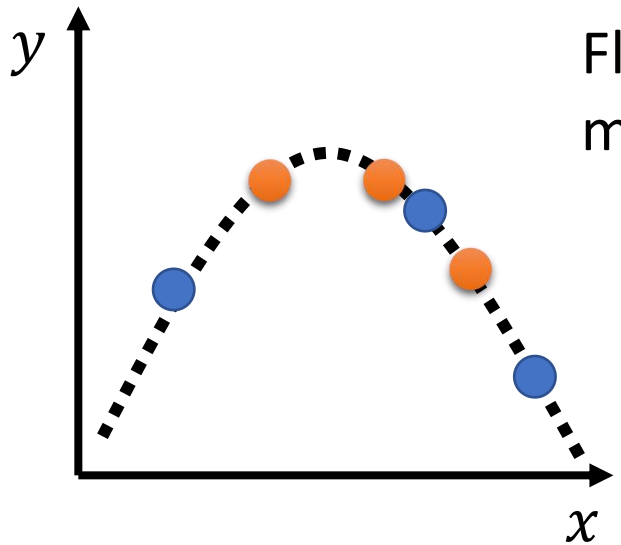
Training data: $\left\{\left(\boldsymbol{x}^{\mathbf{1}}, \hat{y}^{1}\right), \left(\boldsymbol{x}^{\mathbf{2}}, \hat{y}^{2}\right), \ldots, \left(\boldsymbol{x}^{\boldsymbol{N}}, \hat{y}^{N}\right)\right\}$

$$f(\boldsymbol{x}) = \begin{cases} \hat{y}^{i} & \exists \boldsymbol{x}^{\boldsymbol{i}} = \boldsymbol{x} \\ random & otherwise \end{cases}$$

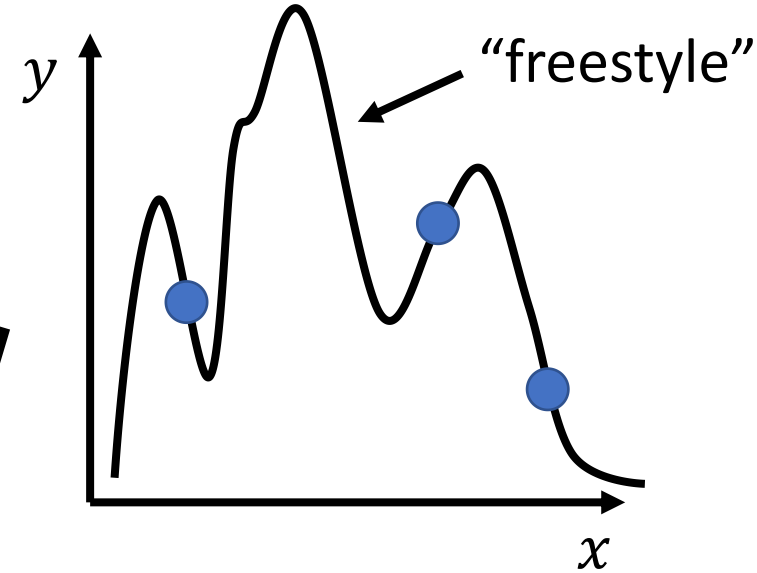
Less than useless …

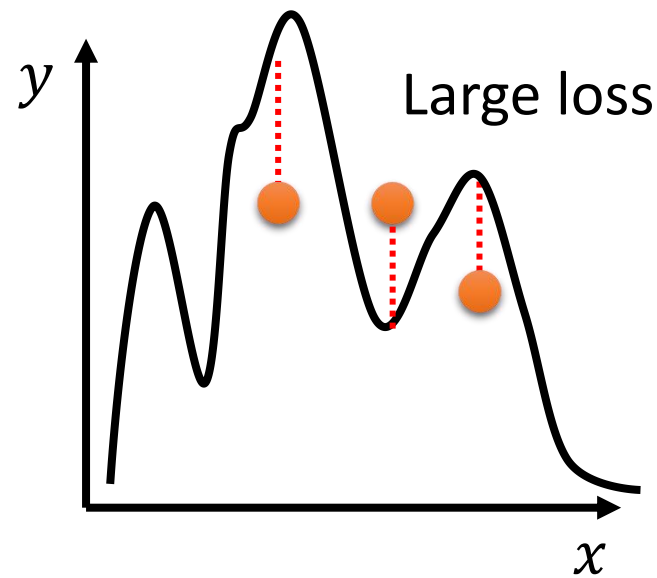
This function obtains **zero training loss**, but **large testing loss**.

Overfitting



Flexible model



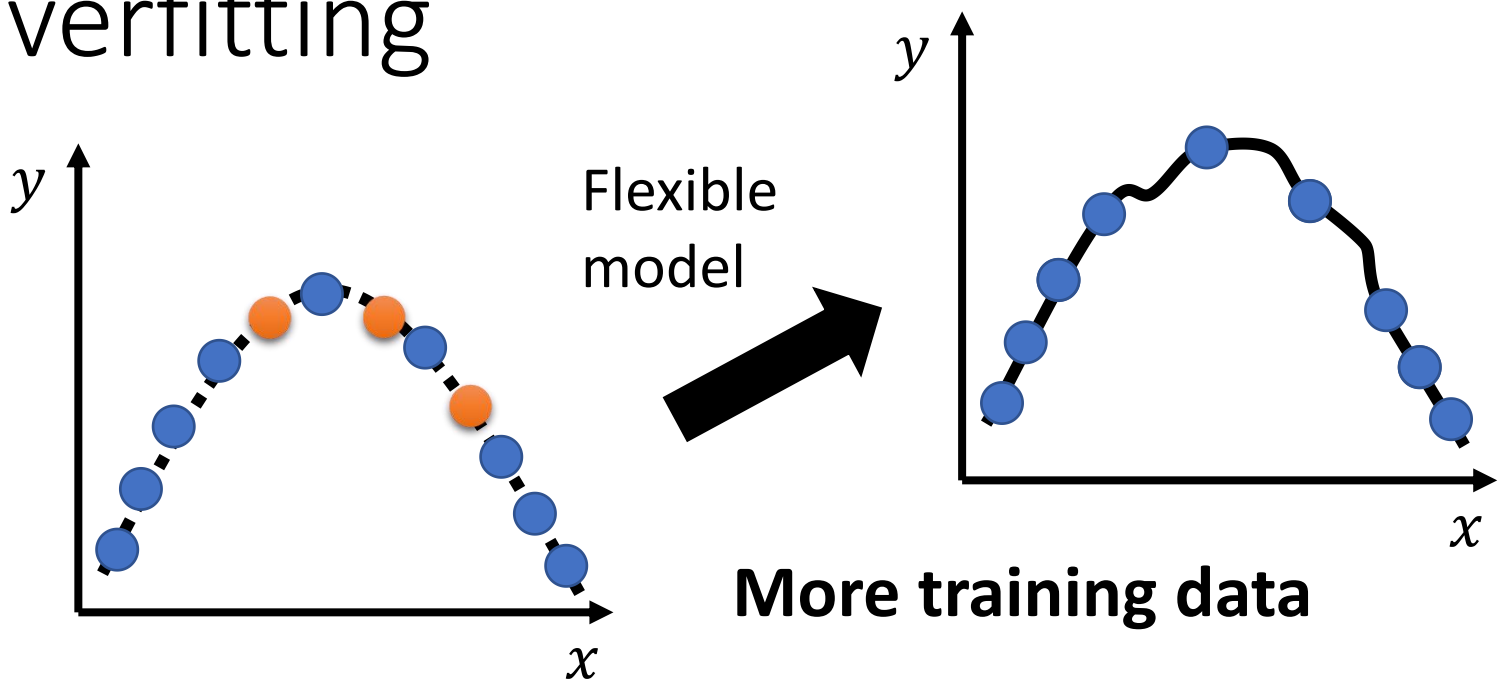


Real data distribution (not observable)

Training data

Testing data

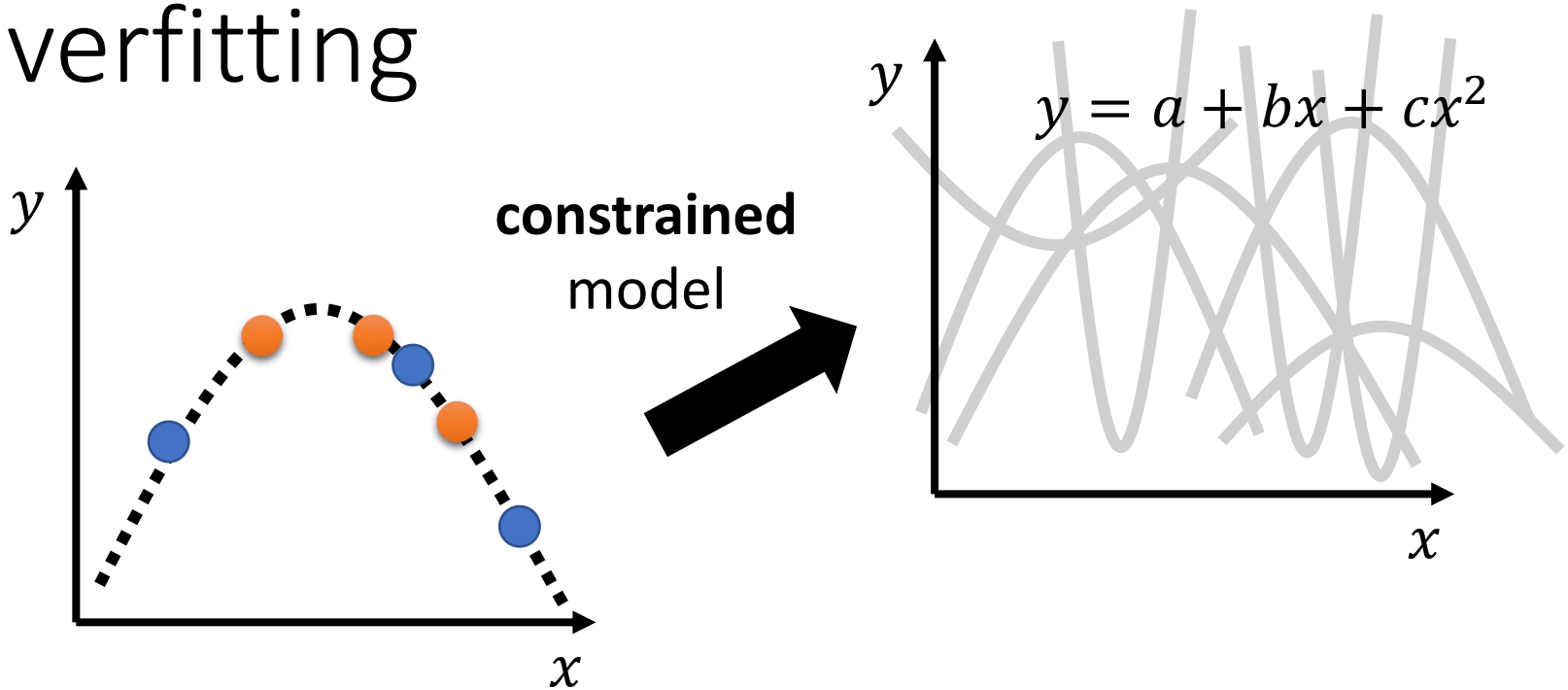
Overfitting



More training data

Data augmentation

Overfitting

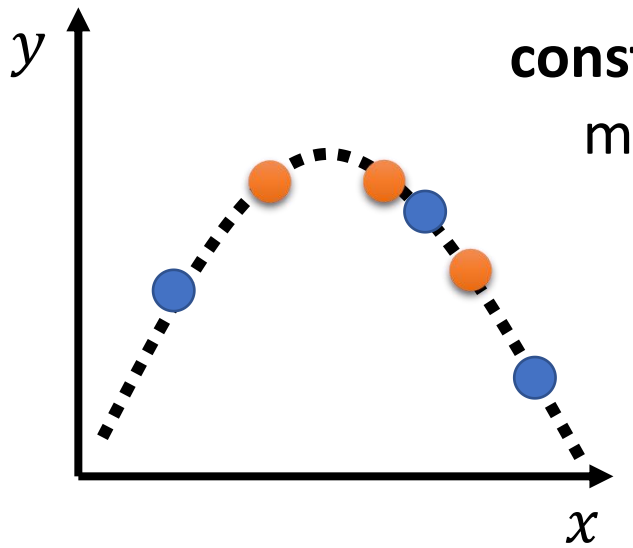


Real data distribution (not observable)

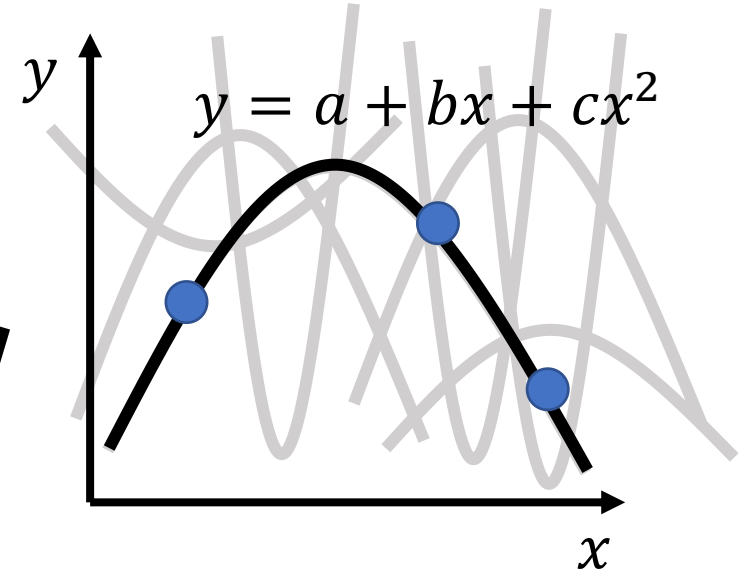
Training data

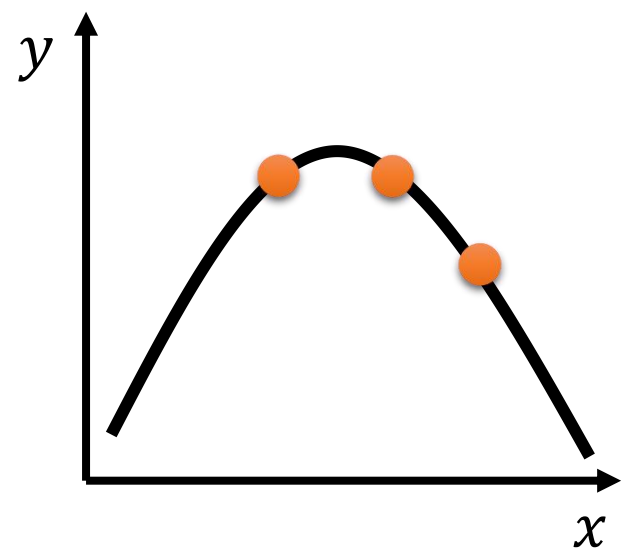
Testing data

Overfitting



constrained model



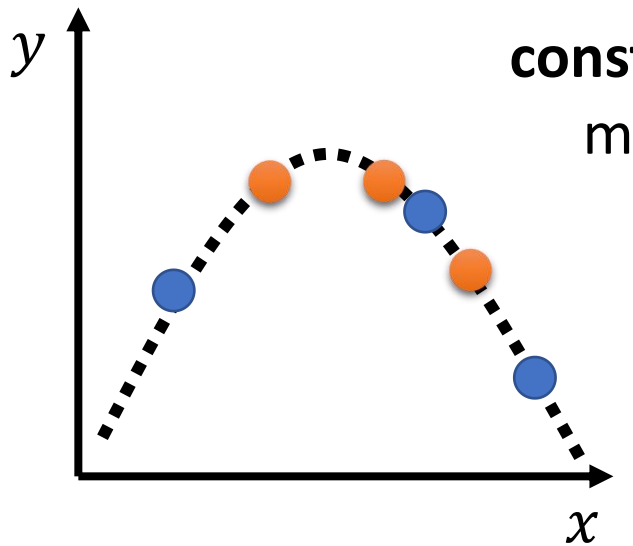


Real data distribution (not observable)

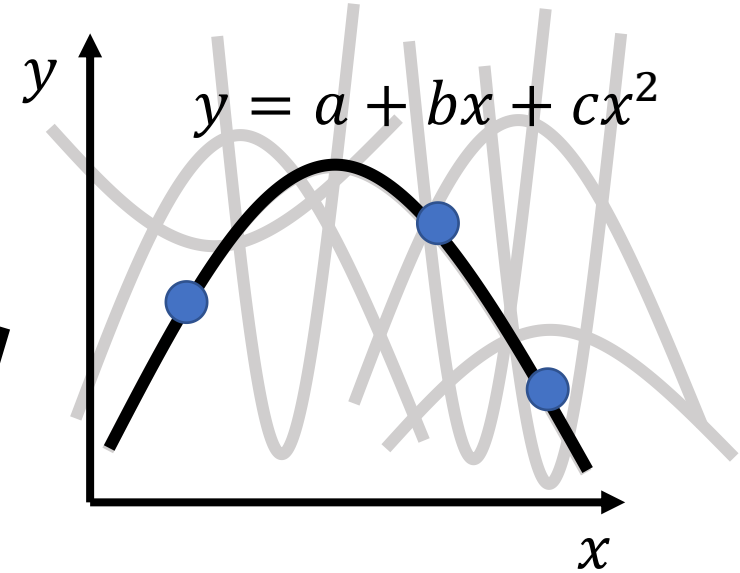
Training data

Testing data

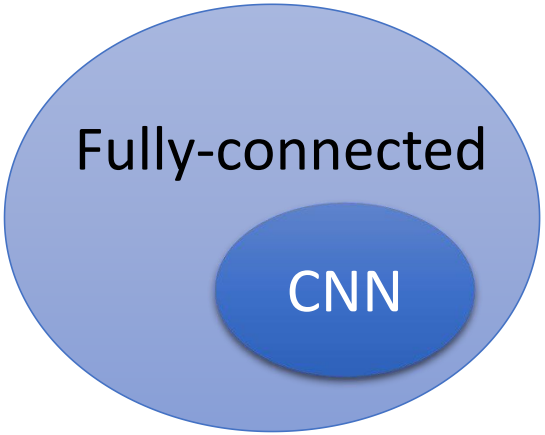
Overfitting



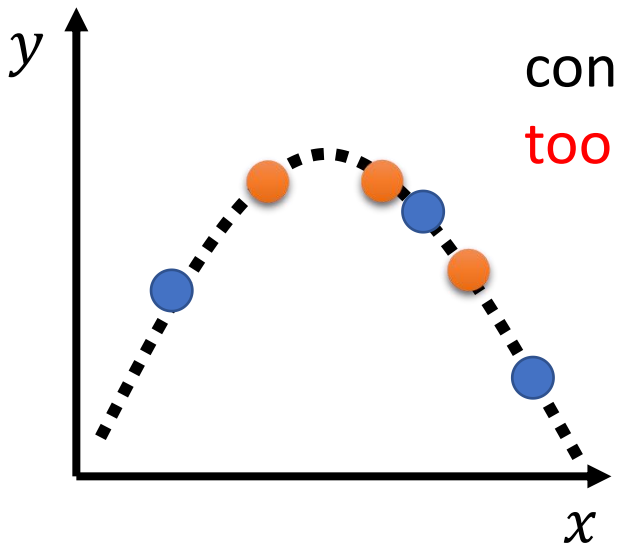
constrained model



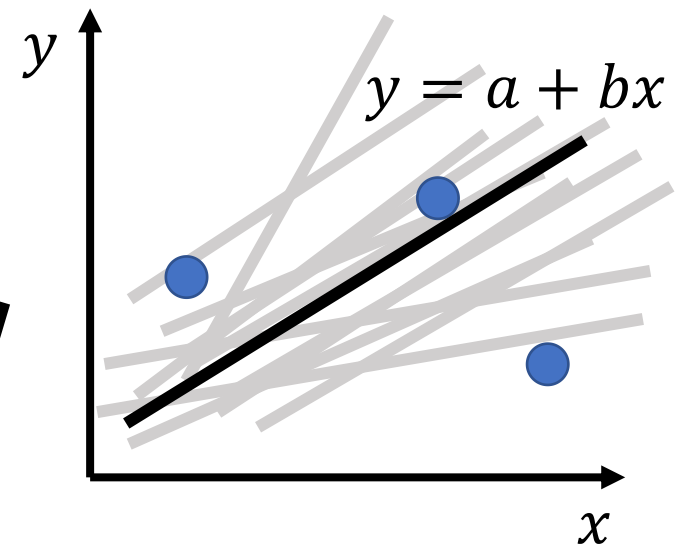
- Less parameters, sharing parameters
- Less features
- Early stopping
- Regularization
- Dropout

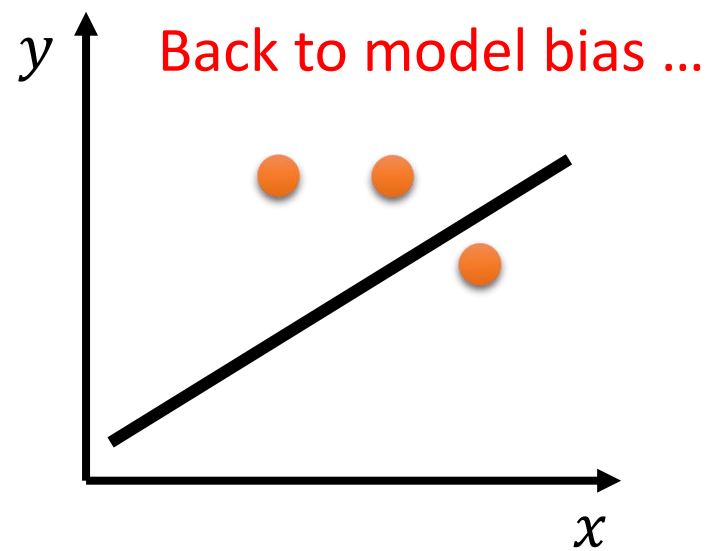


Overfitting



constrain
too much



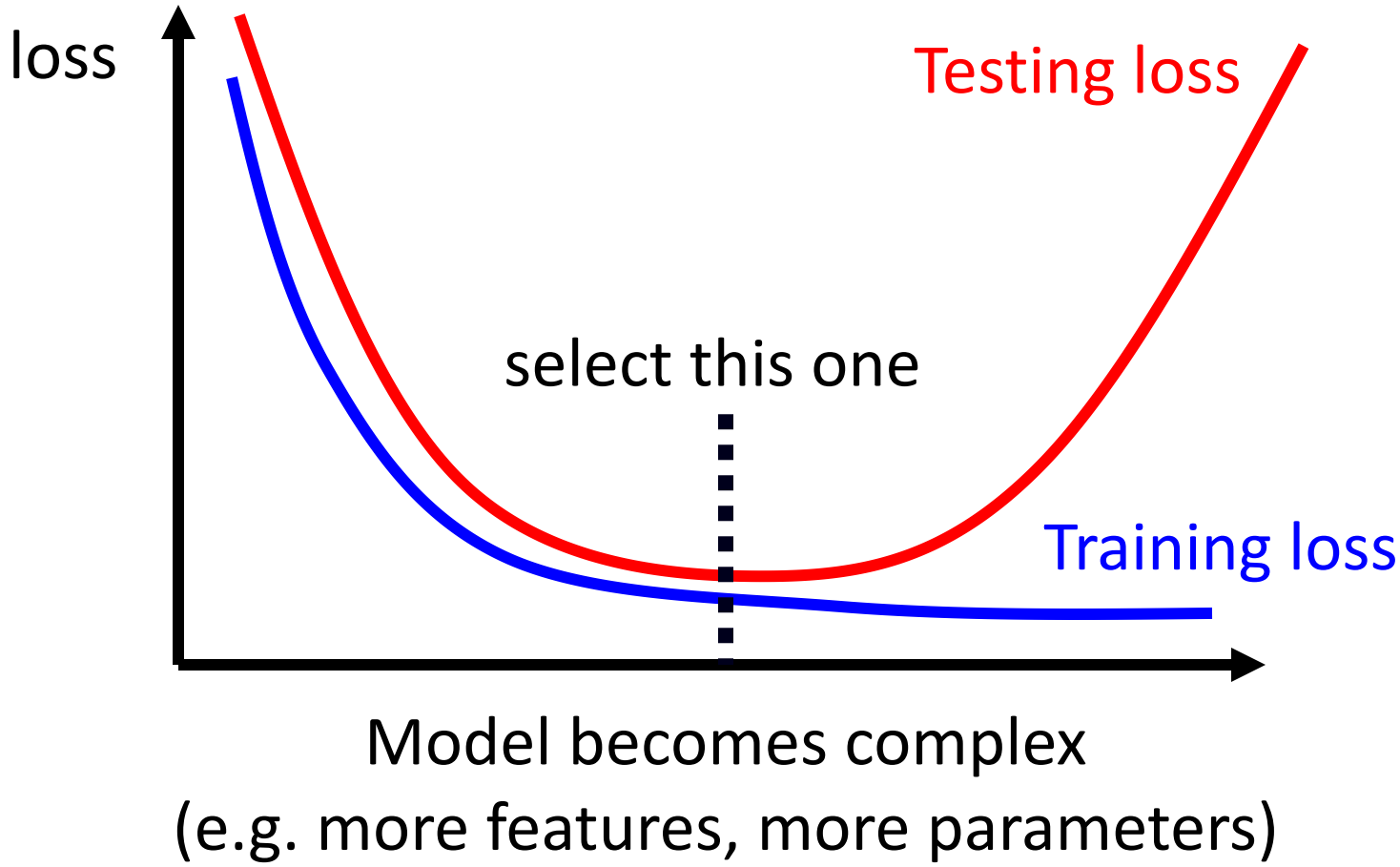


Real data distribution
(not observable)

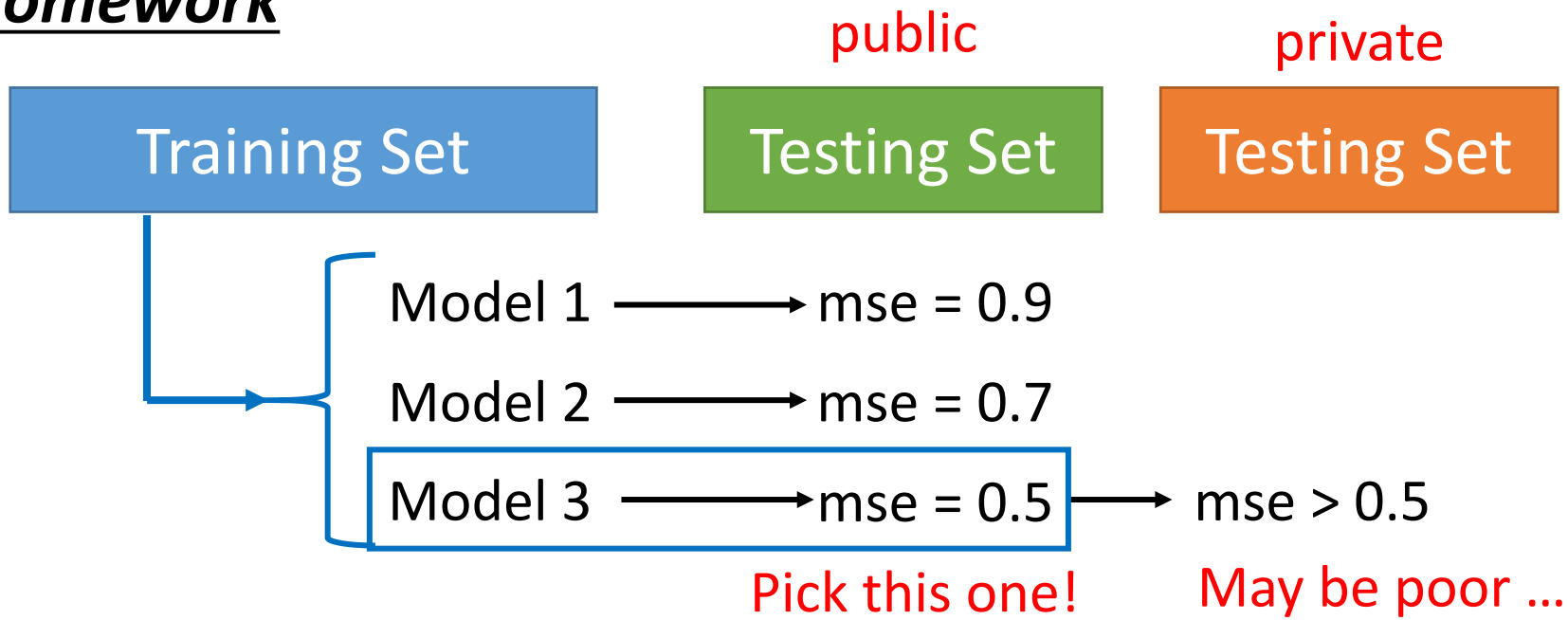
Training data

Testing data

Bias-Complexity Trade-off



Homework



The extreme example again

$$f_k(\boldsymbol{x}) = \begin{cases} \hat{y}^i & \exists \boldsymbol{x}^i = \boldsymbol{x} \\ random & otherwise \end{cases}$$

k: 1 - 1000000000000000000

It is possible that $f_{56789}(\boldsymbol{x})$ **happens** to get good performance on public testing set.

So you select $f_{56789}(\boldsymbol{x})$ …… Random on private testing set

Homework

Training Set	public Testing Set	private Testing Set

Why?

Model 1 → mse = 0.9

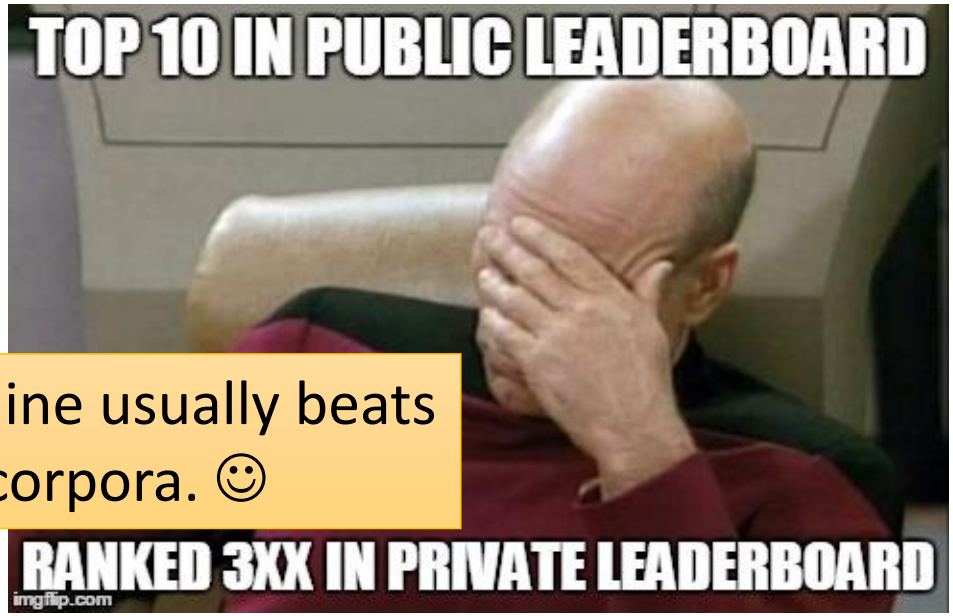
Model 2 → mse = 0.7

Model 3 → mse = 0.5 → mse > 0.5

Pick this one! May be poor …

What will happen?

http://www.chioka.in/how-to-select-your-final-models-in-a-kaggle-competitio/



This explains why machine usually beats human on benchmark corpora. ☺

Cross Validation

How to split?

Training Set

Training Set

Validation set

public

Testing Set

private

Testing Set

Using the results of public testing data to select your model

You are making public set better than private set.

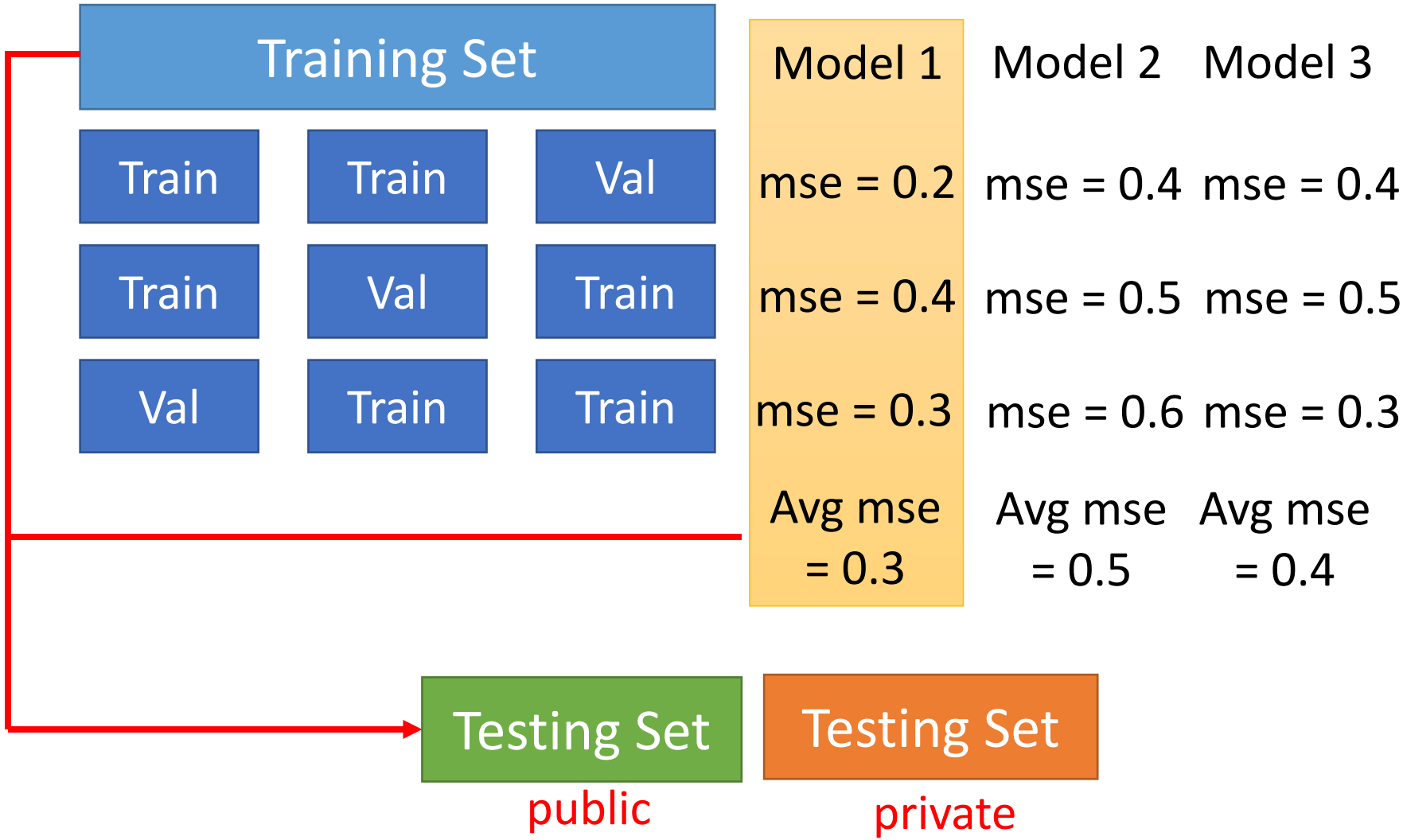
Model 1 → mse = 0.9

Model 2 → mse = 0.7

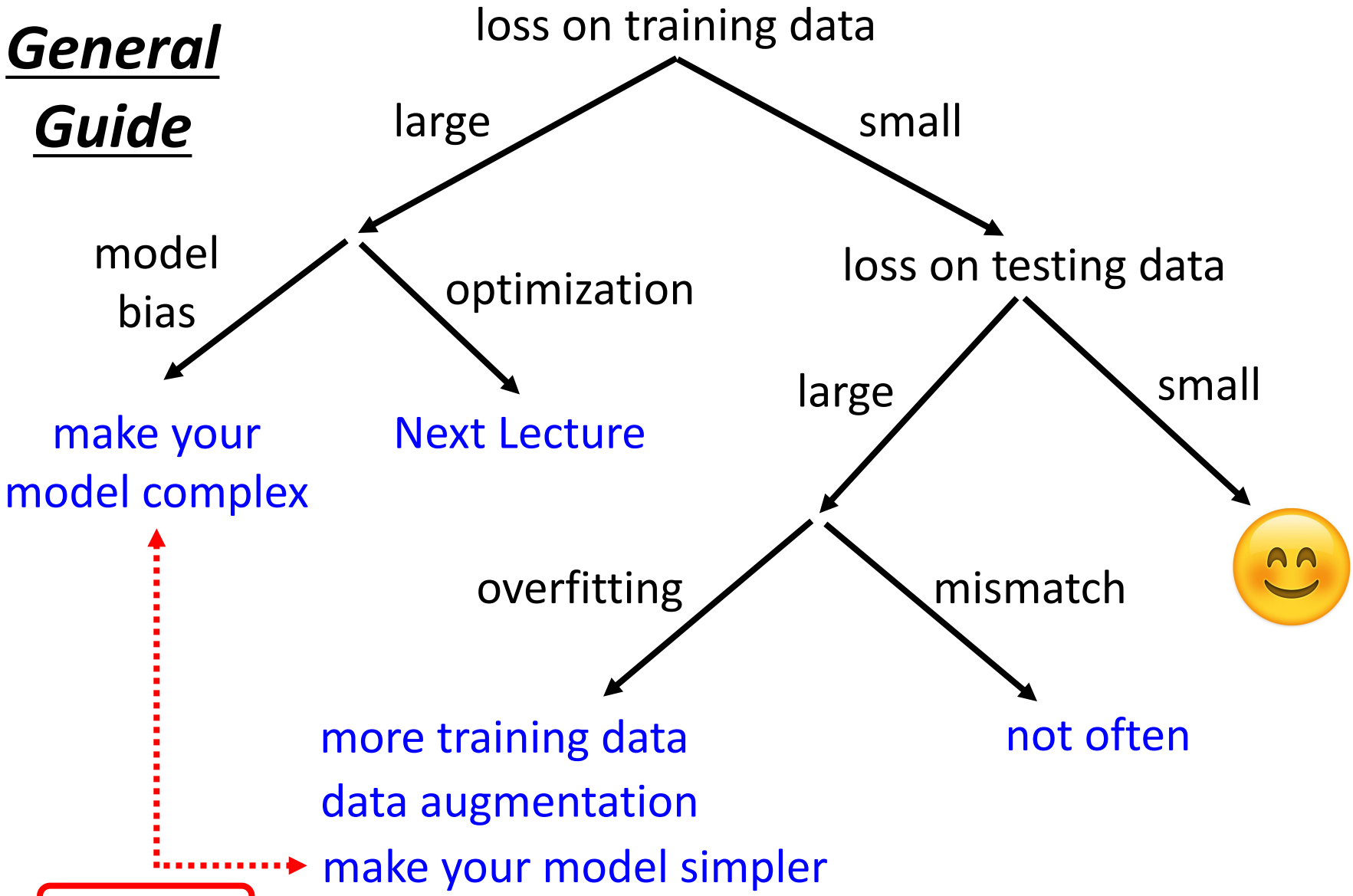
Model 3 → mse = 0.5 → mse > 0.5 → mse > 0.5

Not recommend

N-fold Cross Validation

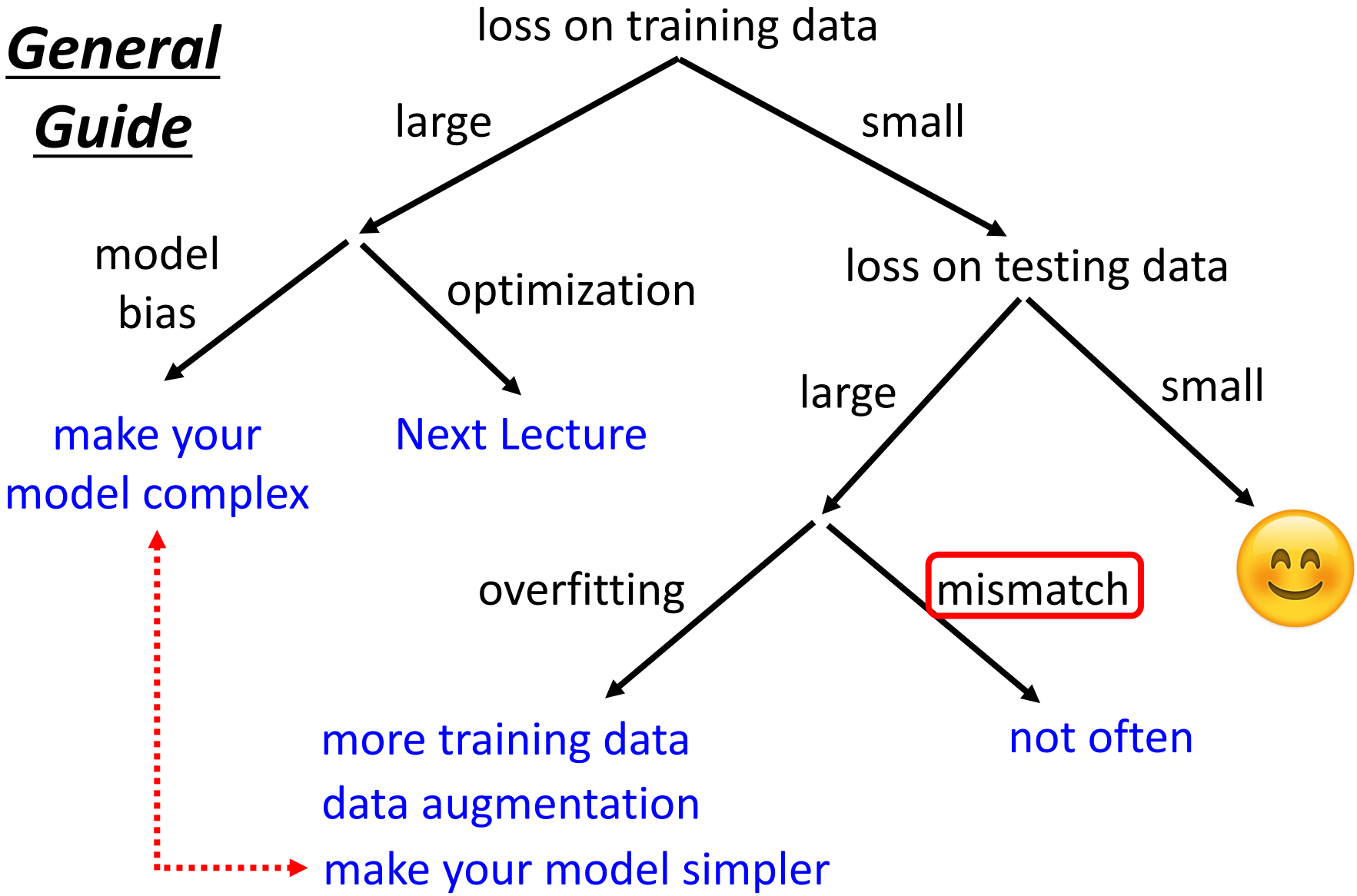


General Guide



Split your training data into training set and validation set for model selection

General Guide



Split your training data into training set and validation set for model selection

Mismatch

- Your training and testing data have different distributions. Be aware of how data is generated.

Training Data

Simply increasing the training data will not help.

Testing Data

General Guide