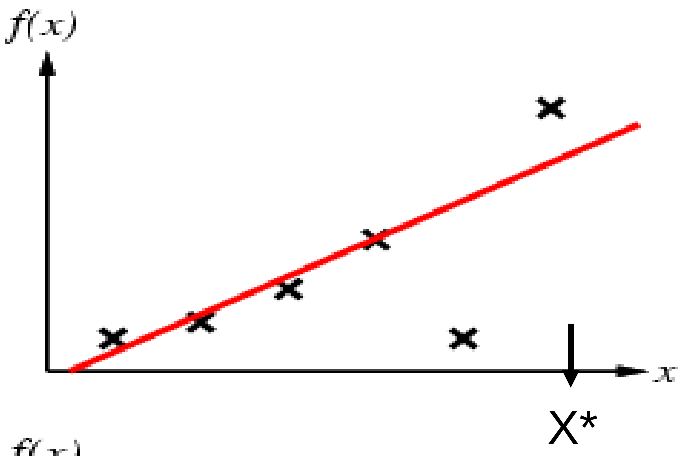
Key Ideas about Learning

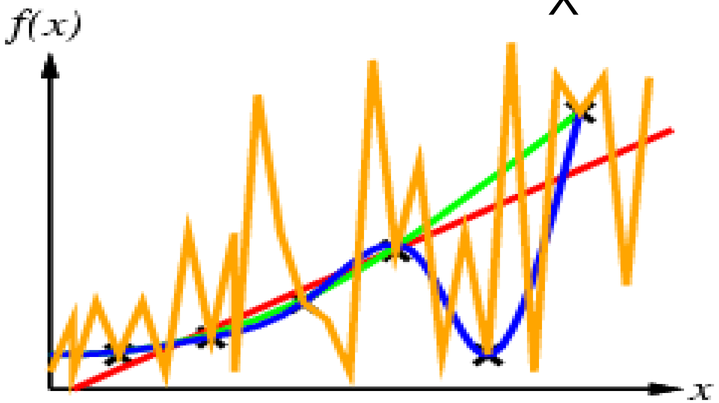
CV Jawahar

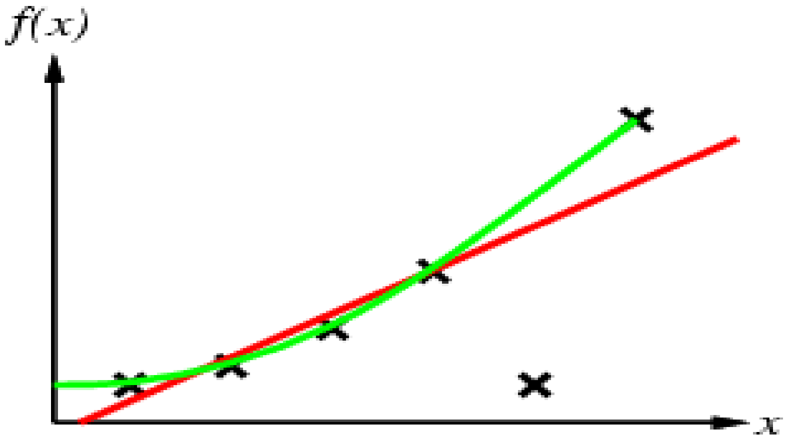
IIIT Hyderabad

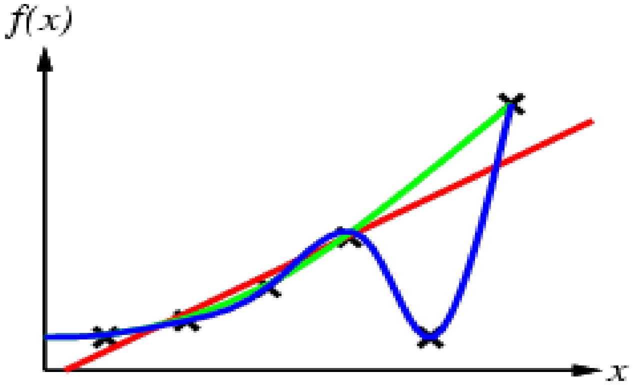
24 Feb, 2025

What is the best model?





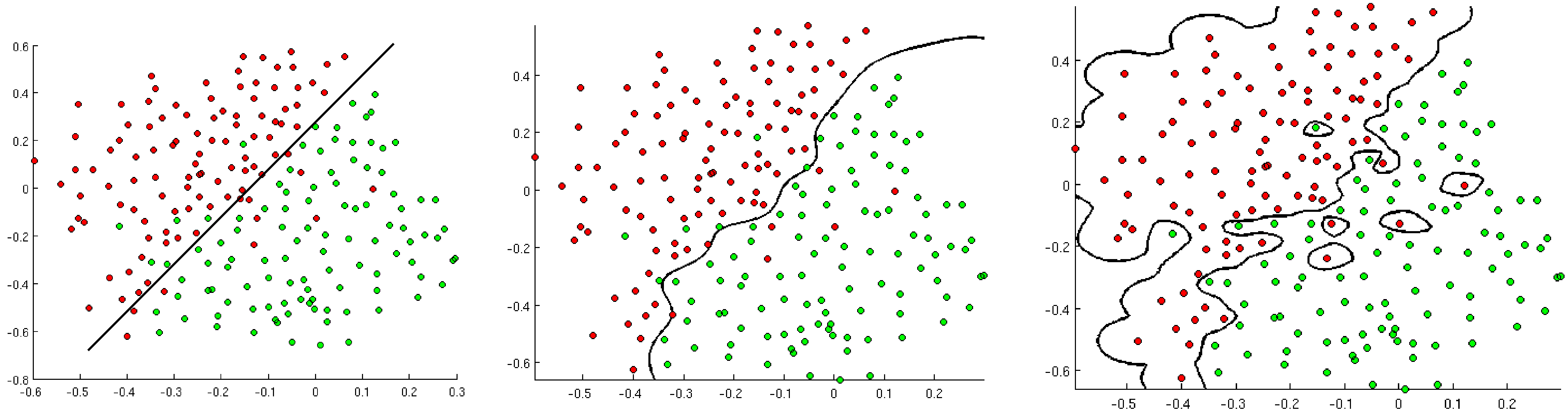




Given six “training” (x_i, y_i) pairs, find the y corresponding to the new “test” x*

Which curve is the best?

Which is the best classifier?



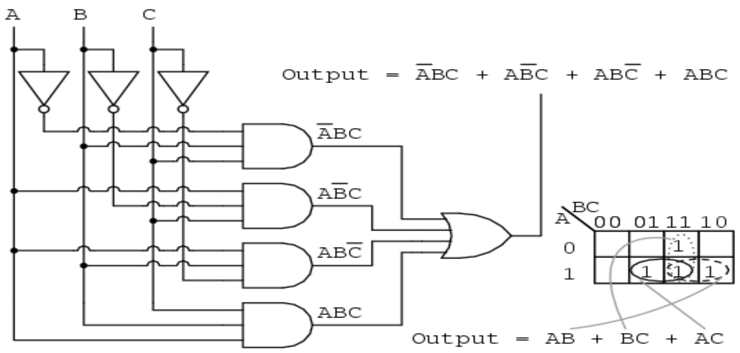
What happens when we simply learn?

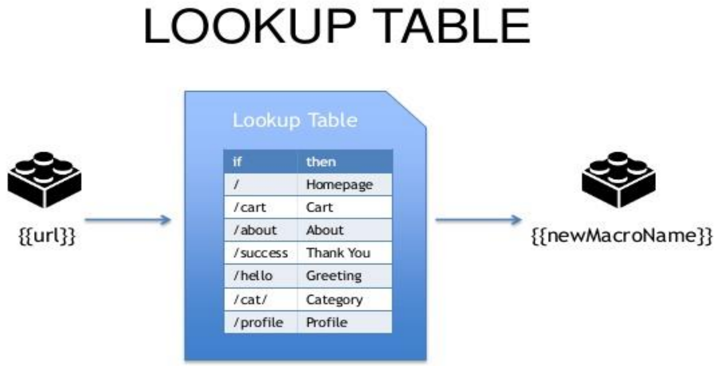
- We often minimize an error over the training examples.
- The discrepancy between the actual and predicted. Let x_i is the input, z_i is the true output and y_i is the predicted output. f() is the model, say a Neural Network.
  - Eg.

$$\sum_i (z_i - y_i)^2 = \sum_i (z_i - f(x_i))^2$$

- If we want only to minimize this, we can even get zero error. The perfect error.
- But we do not want that. **Then what do we want?**

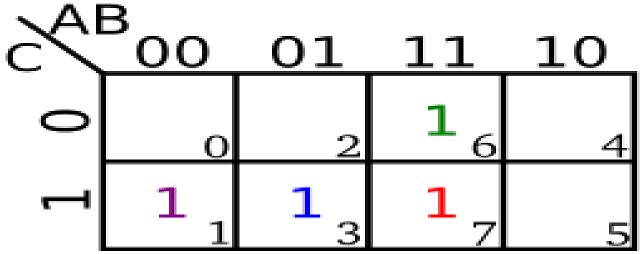
We know about f() that fits samples.

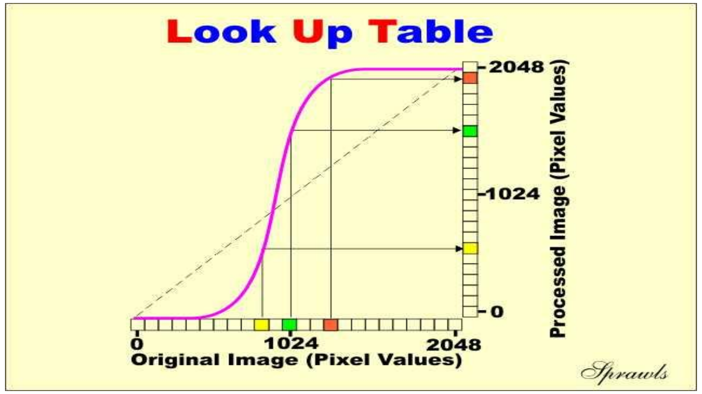




if	then
/	Homepage
/cart	Cart
/about	About
/success	Thank You
/hello	Greeting
/cat/	Category
/profile	Profile

$$f(a,b,c) = \underset{111}{ABC} + \underset{110}{AB\bar{C}} + \underset{011}{\bar{A}BC} + \underset{001}{\overline{AB}C}$$





Is K-Map Learning?

Is LUT Learning?

Generalization

Learning is concerned with accurate prediction of future data, *not* accurate prediction of training or available data.

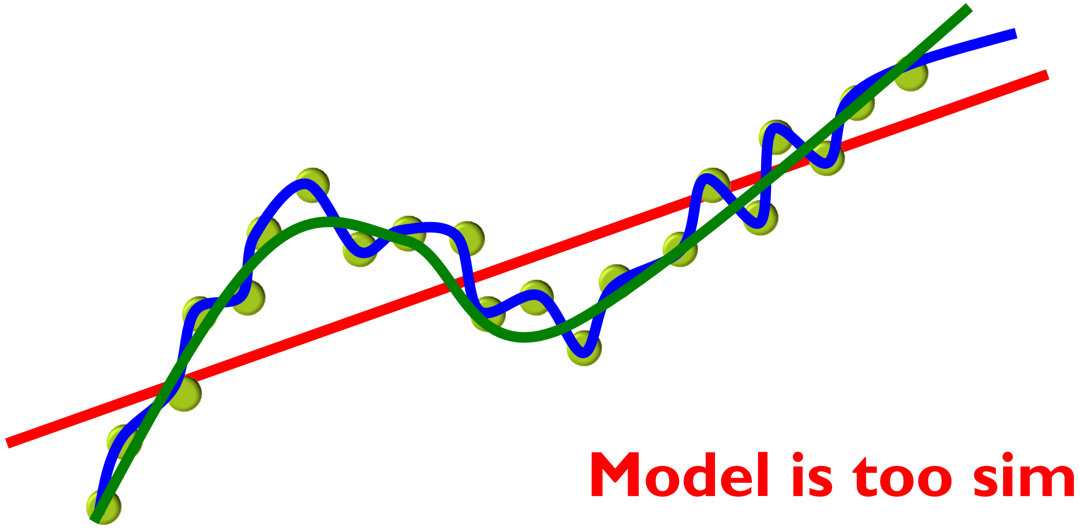
We only have “training data”. Then how do we minimize error on unseen/future data?

Occam’s Razor

Select the simplest hypothesis (solution) that suits the data.

Eg. Minimize Sum of “fit error” and “degree of the polynomial”(complexity of the model)

Model the Signal; Not Noise

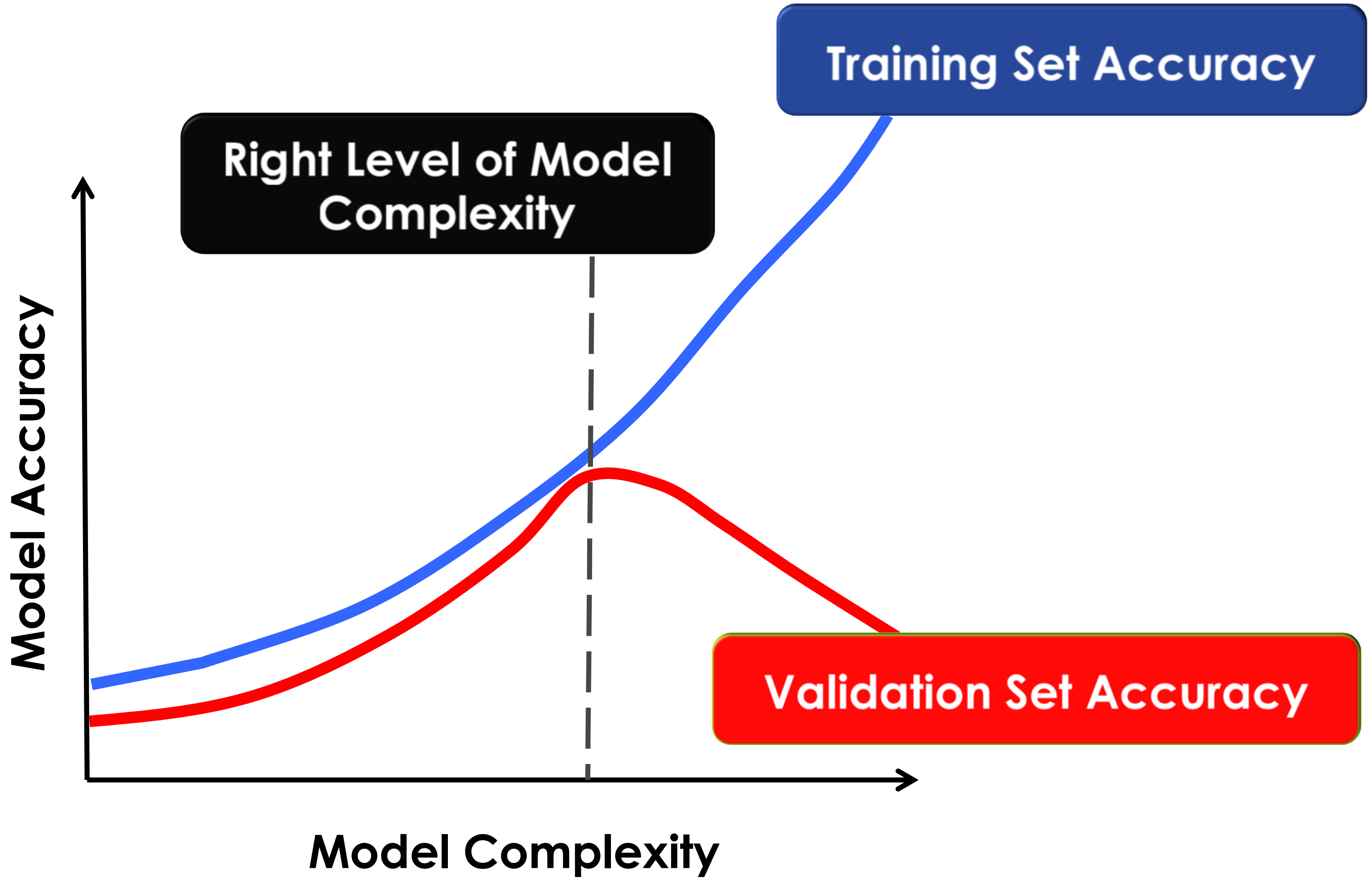


Model is too simple ! UNDER LEARN

Model is too complex ! MEMORIZE

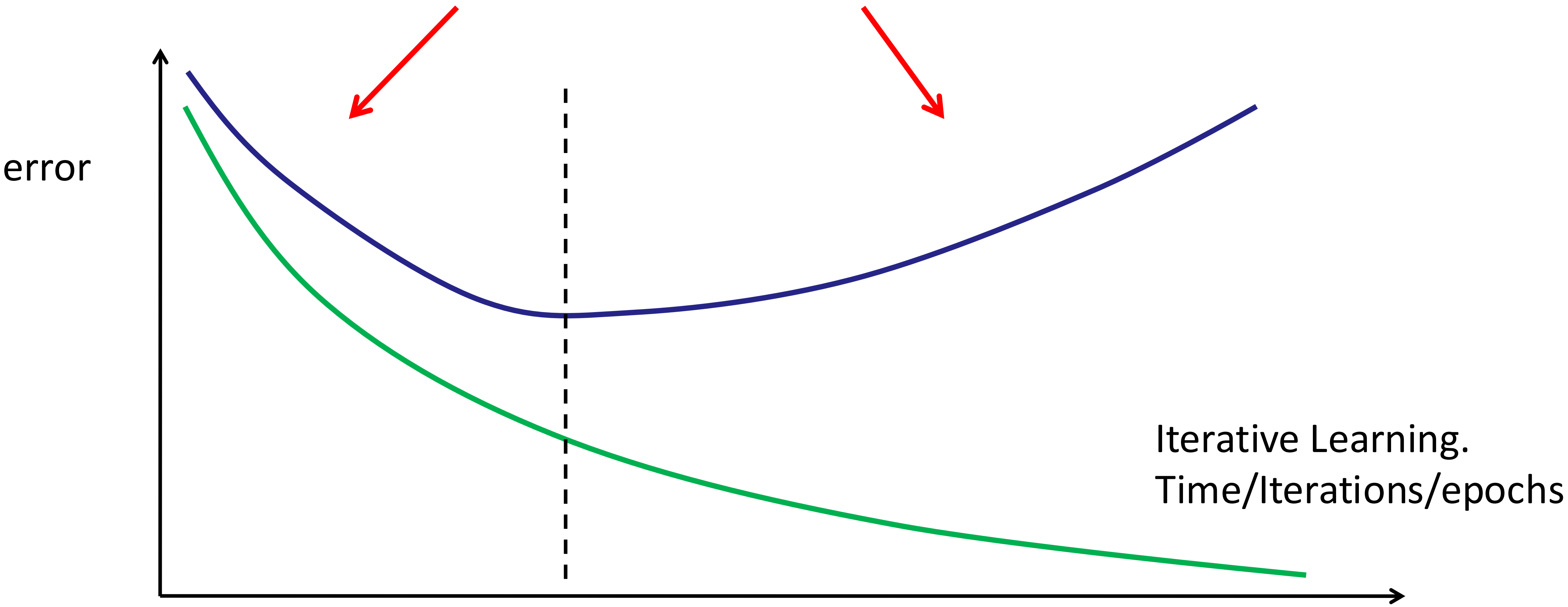
Model is just right ! GENERALIZE

Generalize; Don’t Memorize!



What we may see?

Under-fitting VS. Over-fitting



Overfitting

Overfitting

"Overfitting is a modeling error which occurs when a function is too closely fit to a limited set of data points. Overfitting the model generally takes the form of making an overly complex model to explain idiosyncrasies in the data under study. In reality, the data often studied has some degree of error or random noise within it. Thus, attempting to make the model conform too closely to slightly inaccurate data can infect the model with substantial errors and reduce its predictive power."

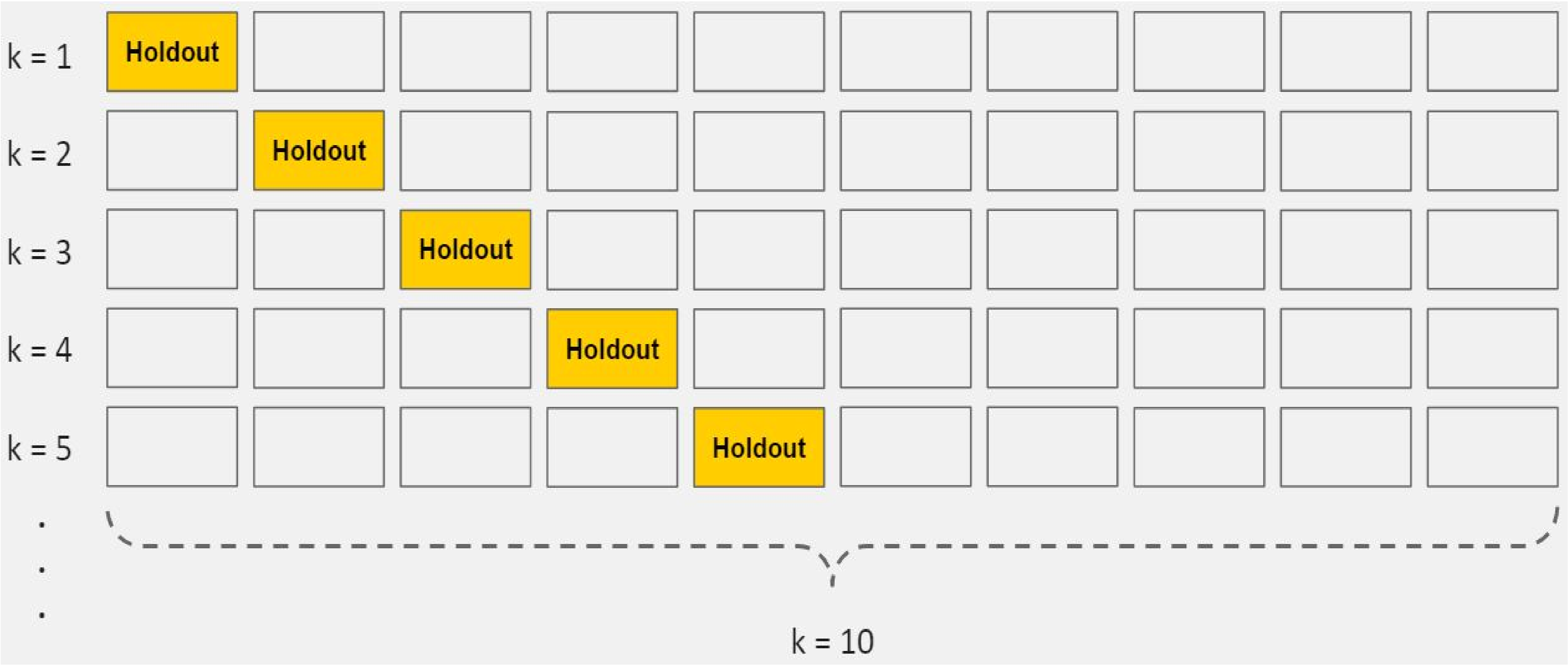
Favorite question ☺: What is the "best" parameter for my training data?

Were you asking for recipe for overfitting? (you know the answer now ☺)…

How to prevent/reduce overfitting?

- Cross Validation
- **Train with more data**
- **Reduce/Remove features**
- Early Stopping
- Regularization
- **Ensembling (may be in a later lecture)**

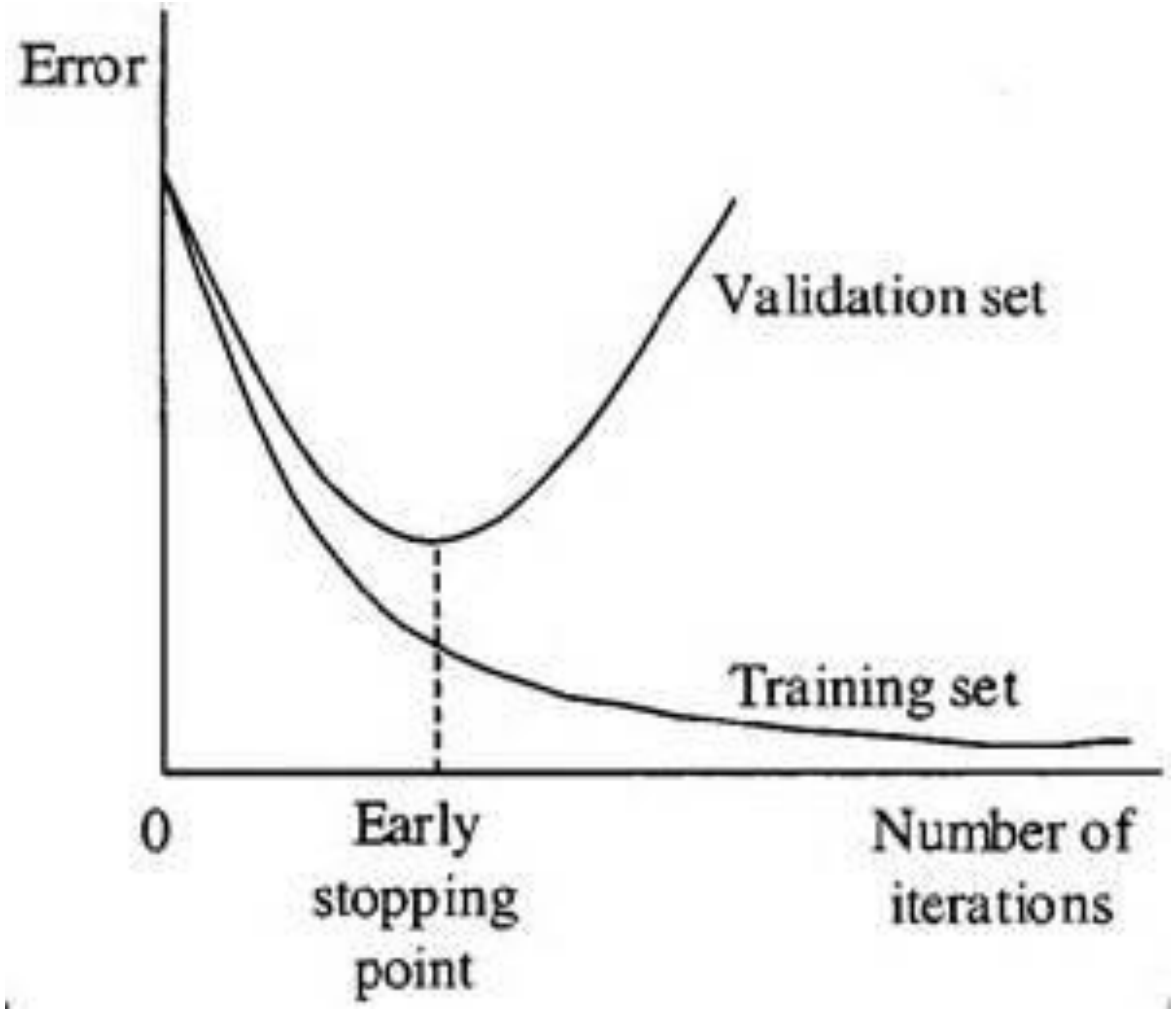
Cross validation



Crossvalidation and LOO

1. Split the training data into k subsets. Use k-1 for training and one for evaluation. Final error is the average of all the k sets.
2. Split the data into x and 100-x percentages for train and evaluation. Repeat the task many times and average the error.
3. If number of samples (N) are small, use N-1 for training and one for testing. Repeat N times and average. Leave One Out (LOO)

Early Stopping



Regularization: **regularization** is a process of introducing additional information in order to solve an ill-posed problem or to prevent overfitting.

What to Minimize? (regularization)

- We were minimizing:

$$\mathcal{E} = \sum_i (z_i - y_i)^2 = \sum_i (z_i - f(x_i))^2$$

- We can minimize
  - Fit error + some measure of complexity of the model

$$\mathcal{E} + \text{ Sum of Weights}$$

$$\mathcal{E} + \text{ Number of Non-zero Weights}$$

Train, Validation, Test Data Sets

- Train: data used for training
  - Find learnable parameters of the model
- Validation: “internal” test data
  - Estimate the error on one or more subsets (eg. Crossvalidation)
  - Vary hyper-parameters (eg. Architecture of a Neural Network) and retrain
- Test
  - Evaluate the final (best) model on test data.

Summary, Questions?

- Find model that is simple and fitting the data.
- Problem of overfitting.
  - How to detect? What to observe during learning? How to avoid.
- Regularize the solution by adding extra term that also minimize the “complexity”.
- Right way to design solution with
  - Train, Val and Test splits.

Questions?