Deep Learning Seminar
Chapter 5. Machine Learning Basics

2017-07-14

Jinwook Kim

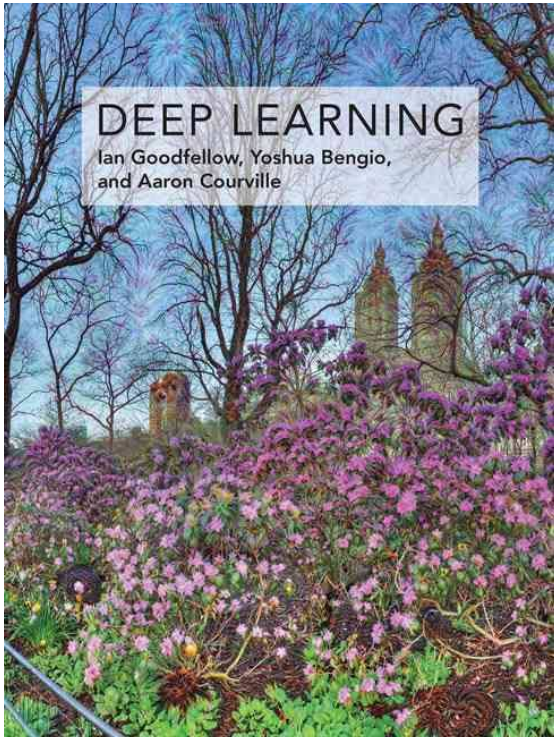
Book Information

- **Title: Deep learning**

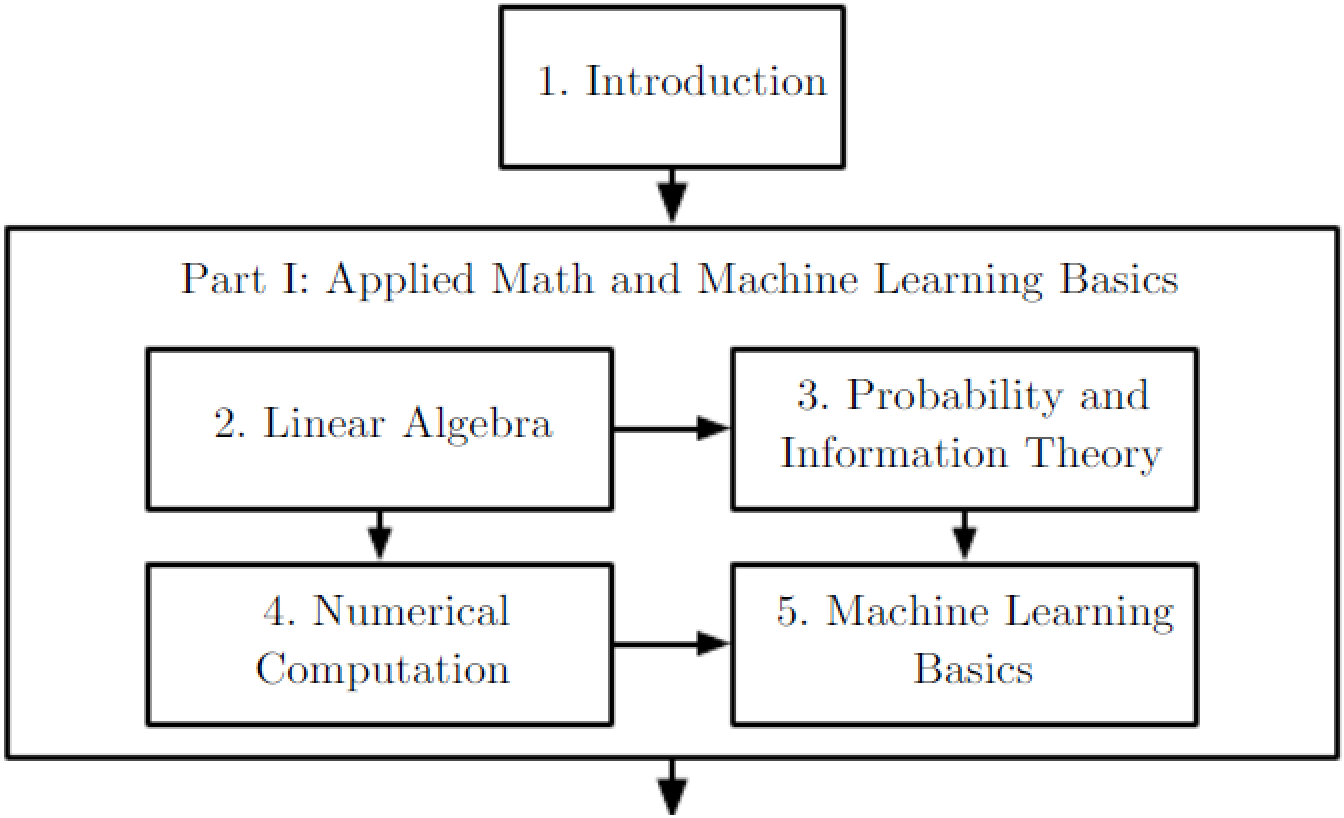
- **Authors:**
  - Ian Goodfellow
  - Yoshua Bengio
  - Aaron Courville

- **Released: November 10, 2016**

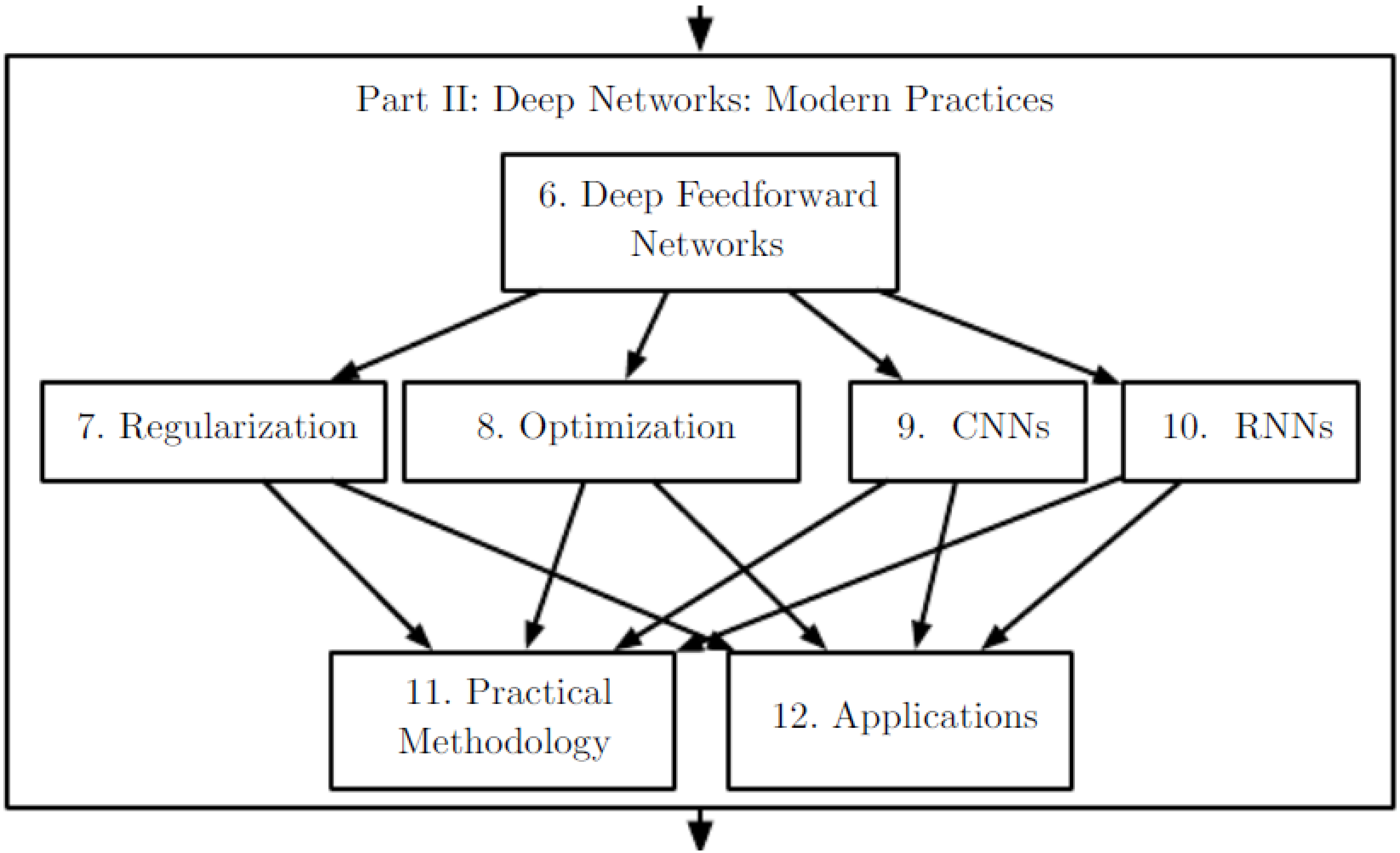
- **ISBN: 9780262337434**



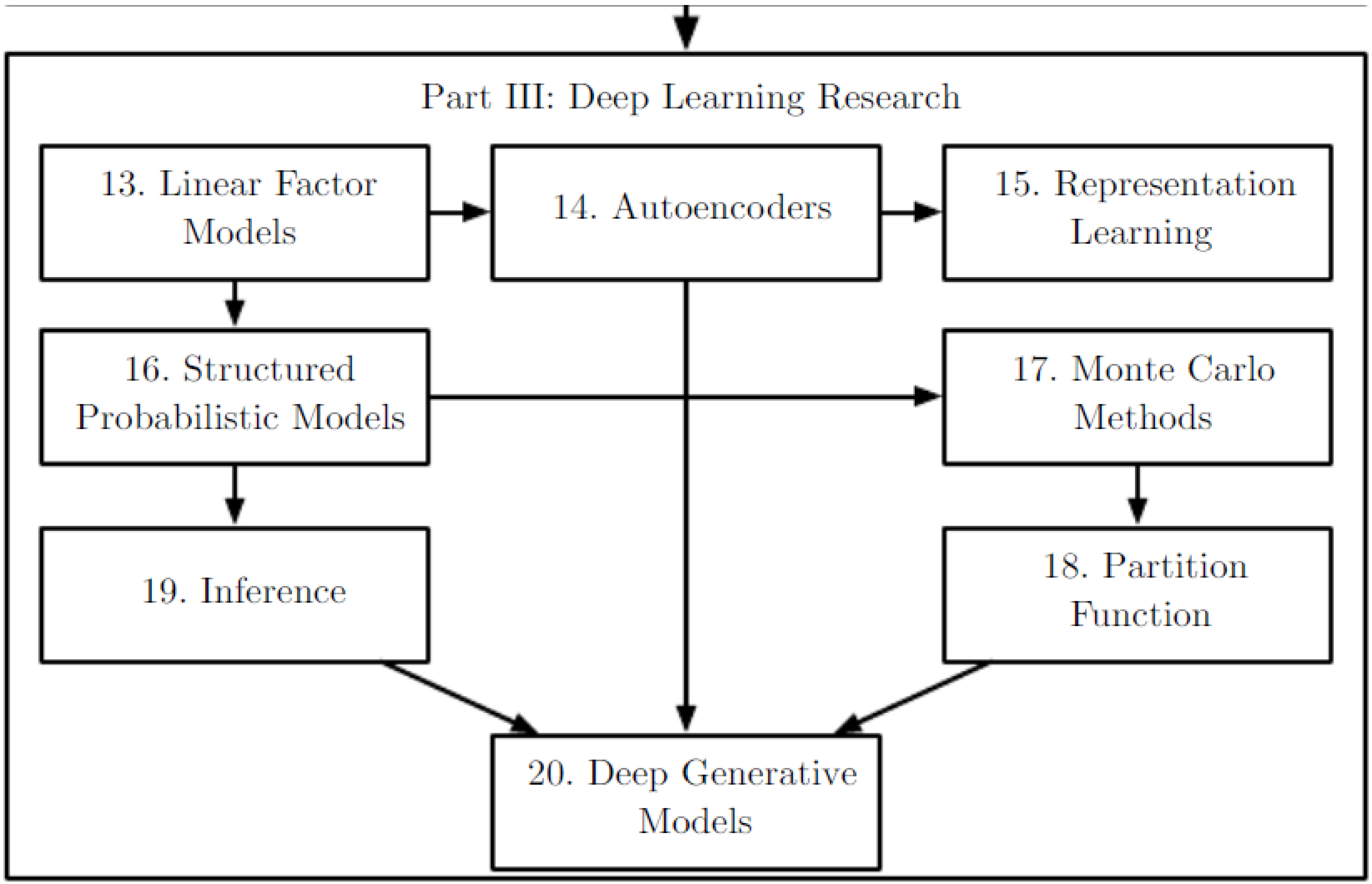
Chapter Organization (Part 1)



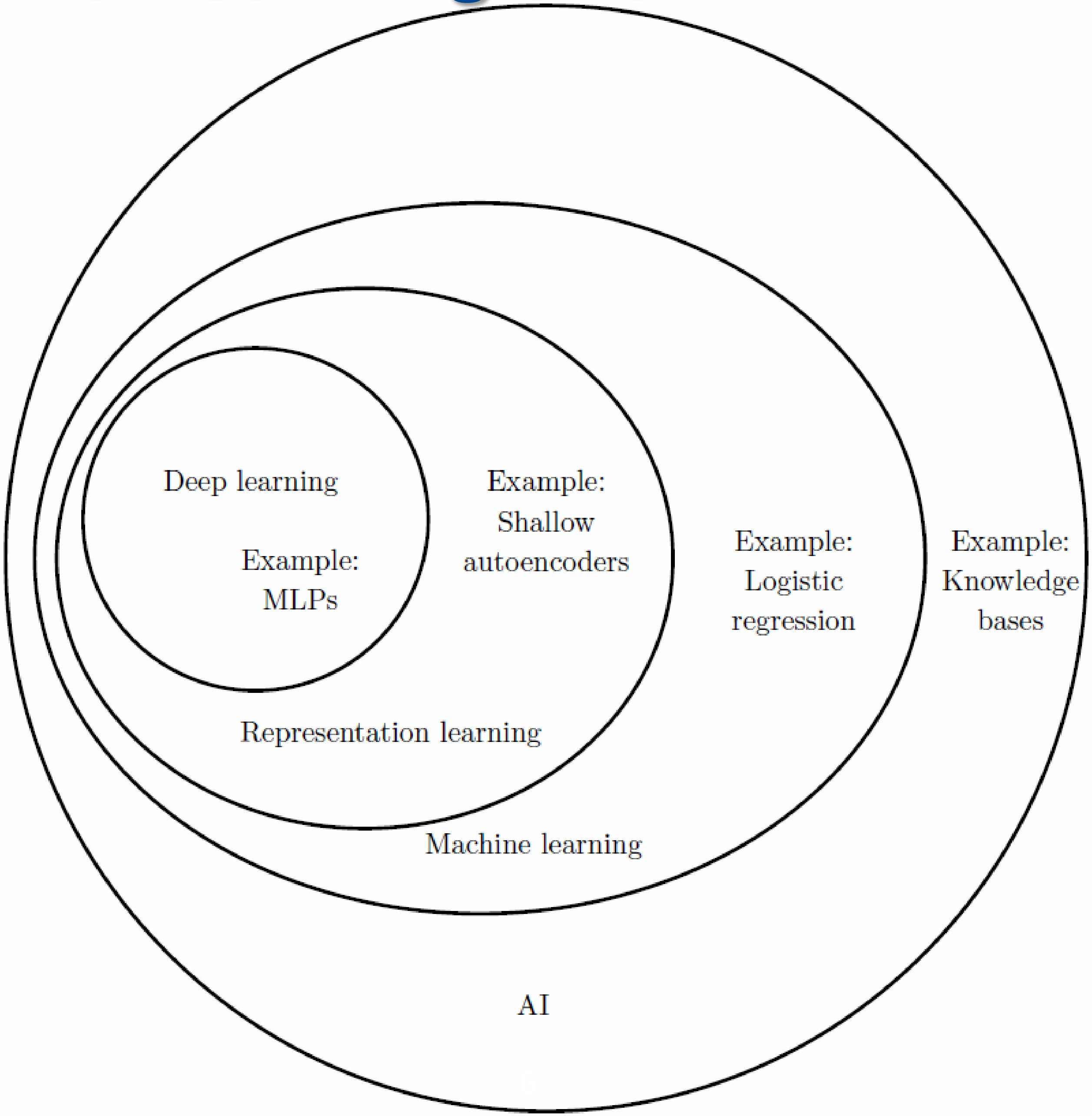
Chapter Organization (Part 2)



Chapter Organization (Part 3)



Machine Learning and AI



Chapter 5. Machine Learning Basics

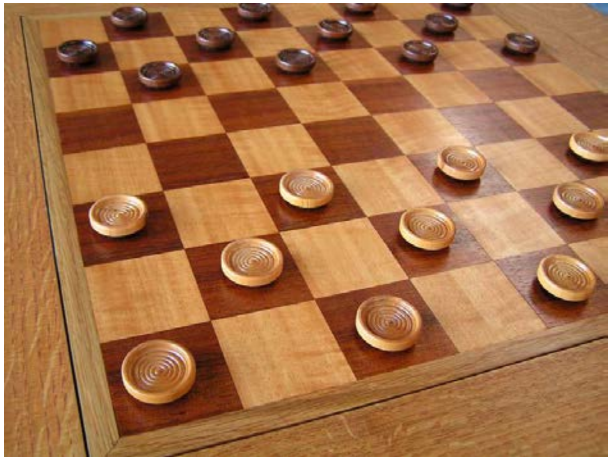
- This chapter provides a brief course in the most important general principles

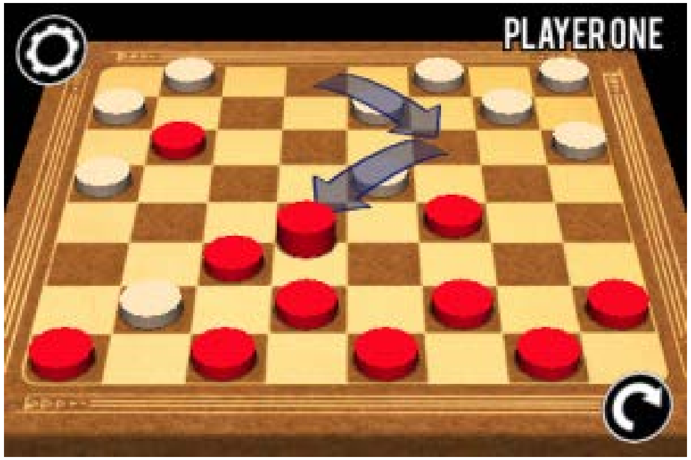
5.1 Learning algorithms
5.2 Capacity, overfitting and underfitting
5.3 Hyperparameters and validation sets
5.4 Estimators, bias and variance
5.5 Maximum likelihood estimation
5.6 Bayesian statistics
5.7 Supervised learning algorithms
5.8 Unsupervised learning algorithms
5.9 Stochastic gradient descent
5.10 Building a machine learning algorithm
5.11 Challenges motivating deep learning

Checkers Game: The First ML Application

- **The Samuel's Checkers-playing program appears to be the world's first self-learning program (Arthur Samuel, 1959)**
  - Over thousands of games, the program started to learn to recognize the patterns, *which patterns led to win or lose*

- **Finally, the program played much better than Samuel himself could**





Learning Algorithms

- **A field of study that gives computers the ability to learn without being *explicitly* programmed (Arthur Samuel, 1959)**

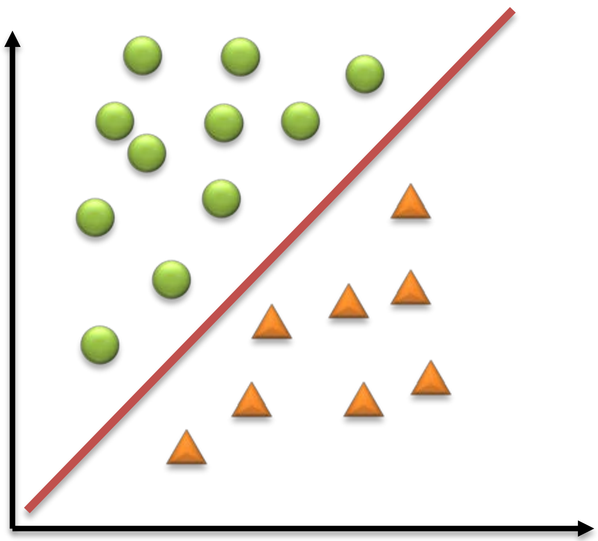
- **A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E (Tom M. Michell, 1997)**
  - E: the experience of playing thousands of games
  - T: playing checkers game
  - P: the fraction of games it wins against human opponents

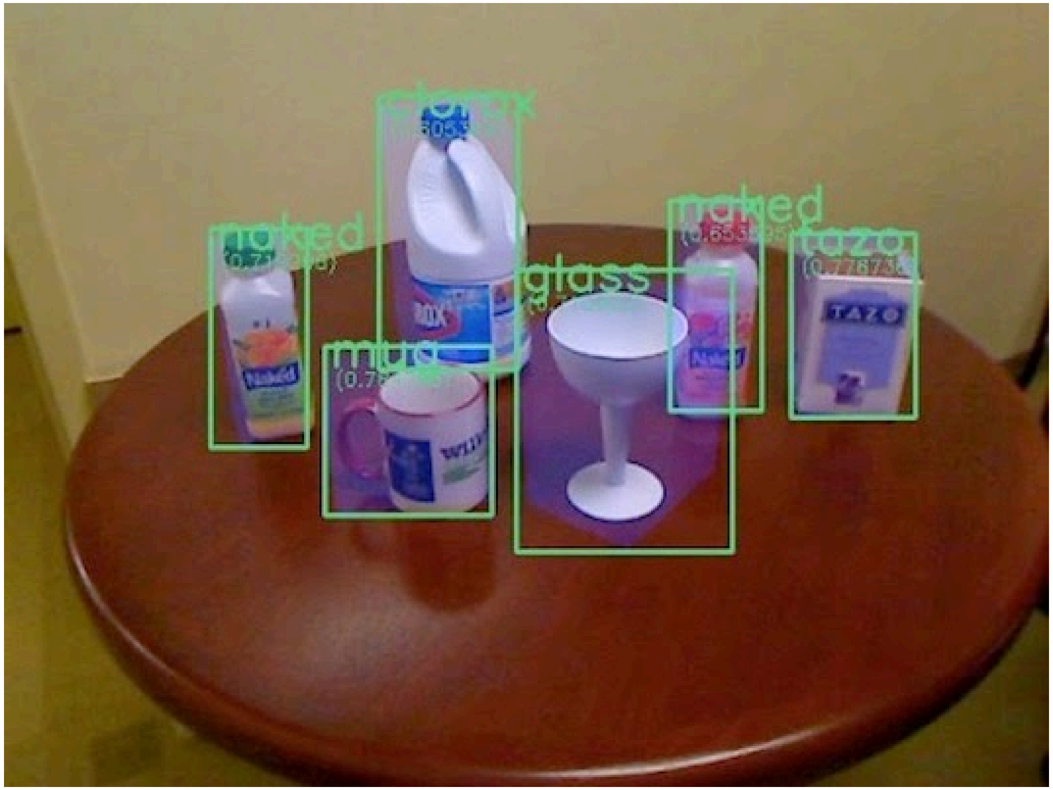
  - By its definition, *Samuel's program has **learned** to play checkers*

Task, T

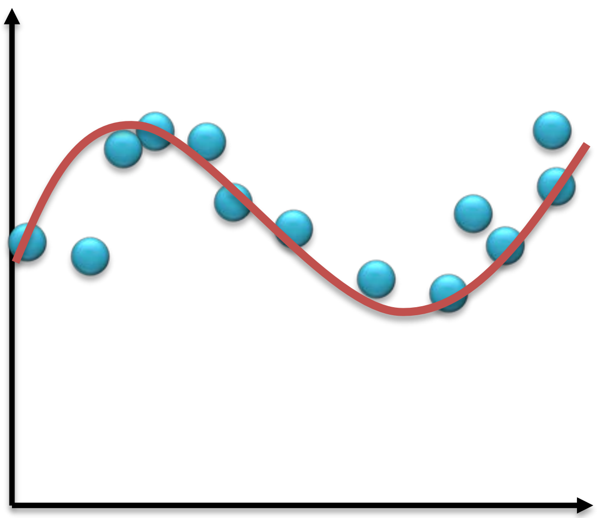
- Classification (with missing inputs)
- Regression
- Transcription
- Machine translation
- Structured output
- Anomaly detection
- Synthesis and sampling
- Imputation of missing values
- Denoising
- Density estimation
- Probability mass function estimation

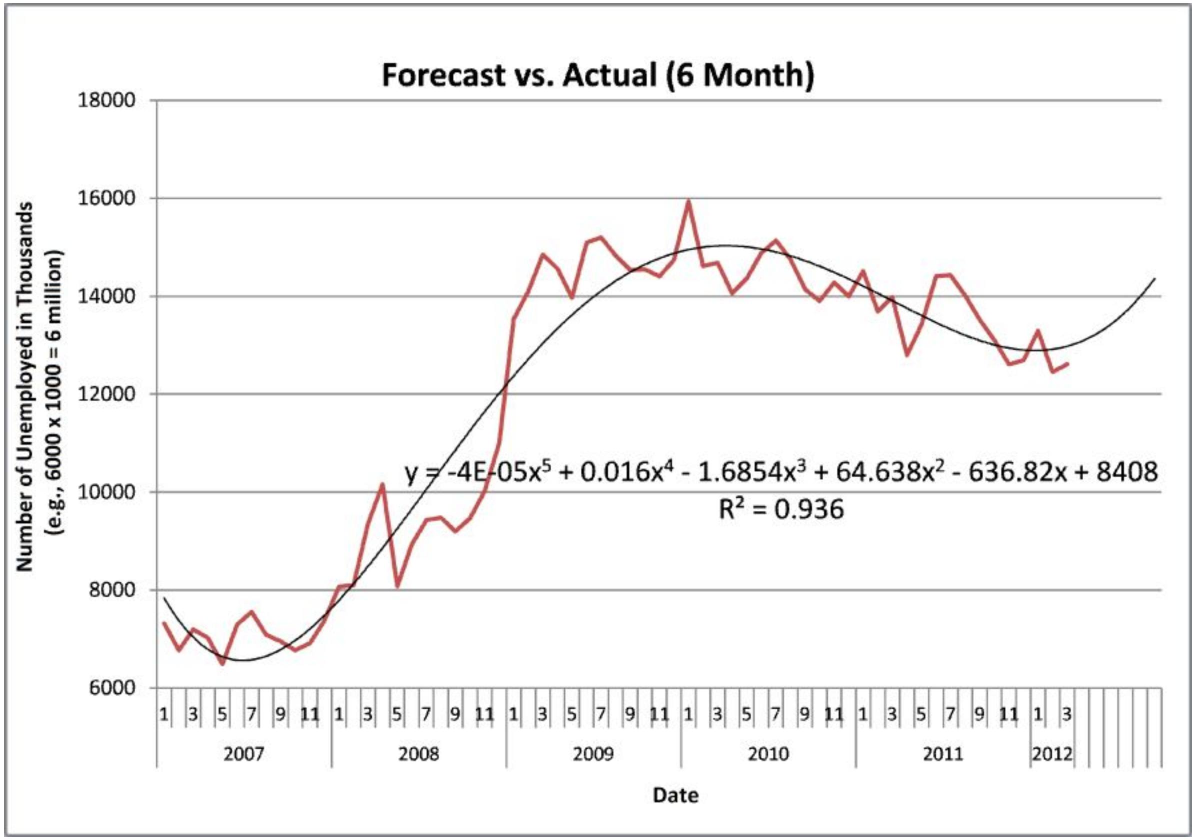
Classification





Regression





Forecast vs. Actual (6 Month)
Number of Unemployed in Thousands (e.g., 6000 x 1000 = 6 million)
18000
16000
14000
12000
10000
8000
6000
y = -4E-05x5 + 0.016x4 - 1.6854x3 + 64.638x2 - 636.82x + 8408
R2 = 0.936
2007
2008
2009
2010
2011
2012
Date

Transcription and Machine Translation

Avenue des Sapins

Performance Measure, P

- **Accuracy (error rate) for categorical data**
  - To measure the proportion of examples for which model produces the correct output

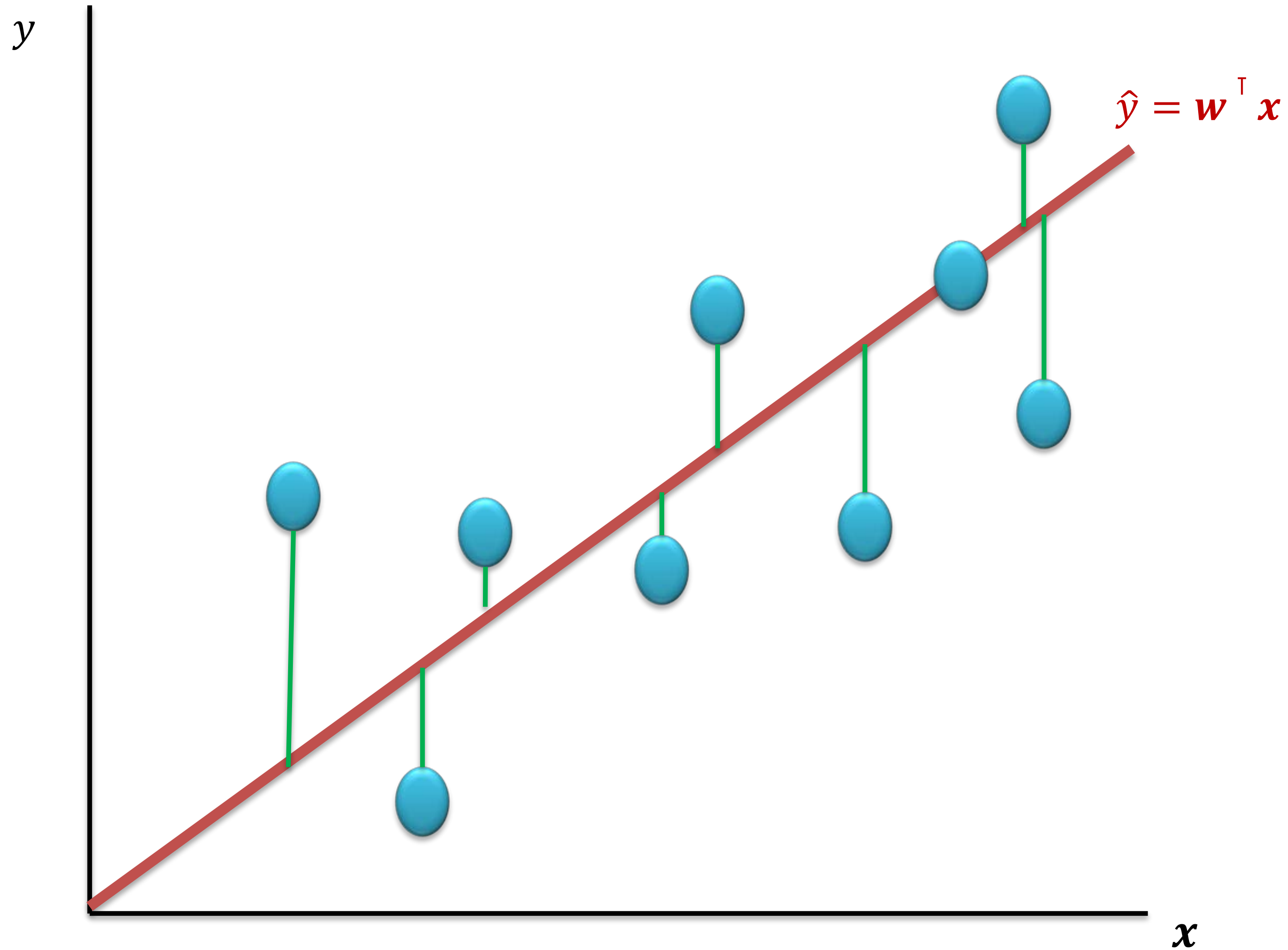
- **Average log-probability for density estimation**
  - To measure continuous-valued score for each example

- **E.g., test set for validating model with unseen data**
  1. Separate the data into training set and test set
  2. Train the model with training set
  3. Measure the model's performance with test set

Experience, E

- **Unsupervised learning algorithms,**
  - Experiences a dataset containing many features
  - Learns useful properties of the structure of the dataset
- **Supervised learning algorithms,**
  - Experiences a dataset associated with labels
  - Learns to predict the labels from the data
- **Reinforcement learning algorithms,**
  - Not just experience with a fixed dataset,
    but interact with an environment
  - Learns actions to maximize cumulative rewards

Example: Linear Regression



Formal Definition of Linear Regression

- **Task, T: to predict y from x by outputting $\hat{y} = \boldsymbol{w}^{\top}\boldsymbol{x}$**
  - $\boldsymbol{x} \in \mathbb{R}^n$: input data
  - $y \in \mathbb{R}$: output value ($\hat{y}$: predicted by the model)
  - $\boldsymbol{w} \in \mathbb{R}^n$: parameters (or weights)

- **Experience, E: training set $(\boldsymbol{X}^{\boldsymbol{train}}, \boldsymbol{y}^{\boldsymbol{train}})$**

- **Performance measure, P:
mean squared error (MSE) on $(\boldsymbol{X}^{\boldsymbol{test}}, \boldsymbol{y}^{\boldsymbol{test}})$**

$$\boldsymbol{MSE_{test}} = \frac{1}{m}\sum_{i}(\hat{y}^{\,\boldsymbol{test}} - y^{\,\boldsymbol{test}})_i^2$$

m: number of dataset

Find w by Minimizing MSE_{train}

$$\nabla w MSE_{train} = 0$$

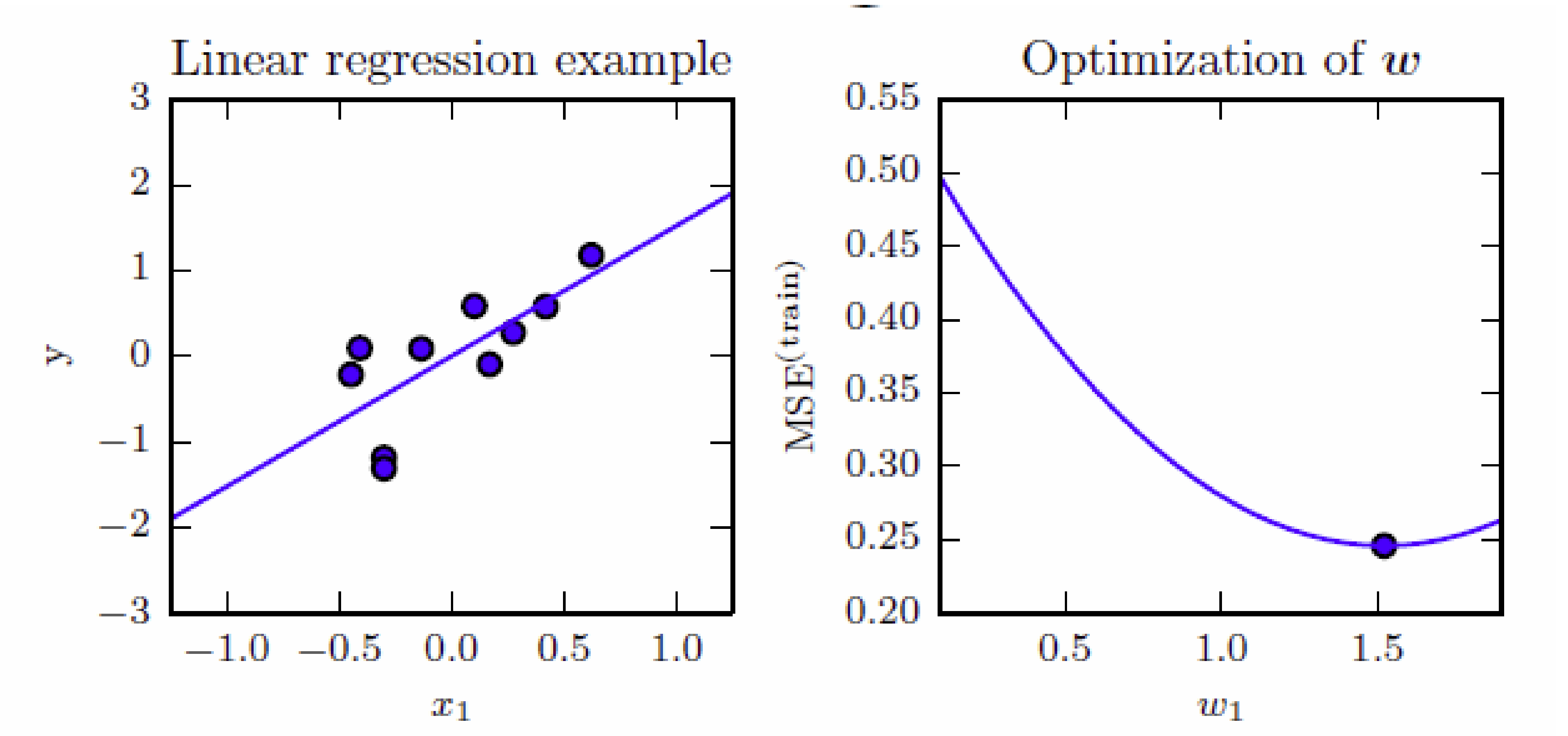
$$\Rightarrow \nabla w MSE_{train} = \frac{1}{m} \sum_i (\hat{y}^{train} - y^{train})_i^2 = 0$$

$$\Rightarrow \nabla w MSE_{train} = \frac{1}{m} \sum_i (X^{train} w - y^{train})_i^2 = 0$$

…

$$\Rightarrow w = (X^{train^{\top}} X^{train})^{-1} X^{train^{\top}} y^{train}$$

Linear Regression Problem



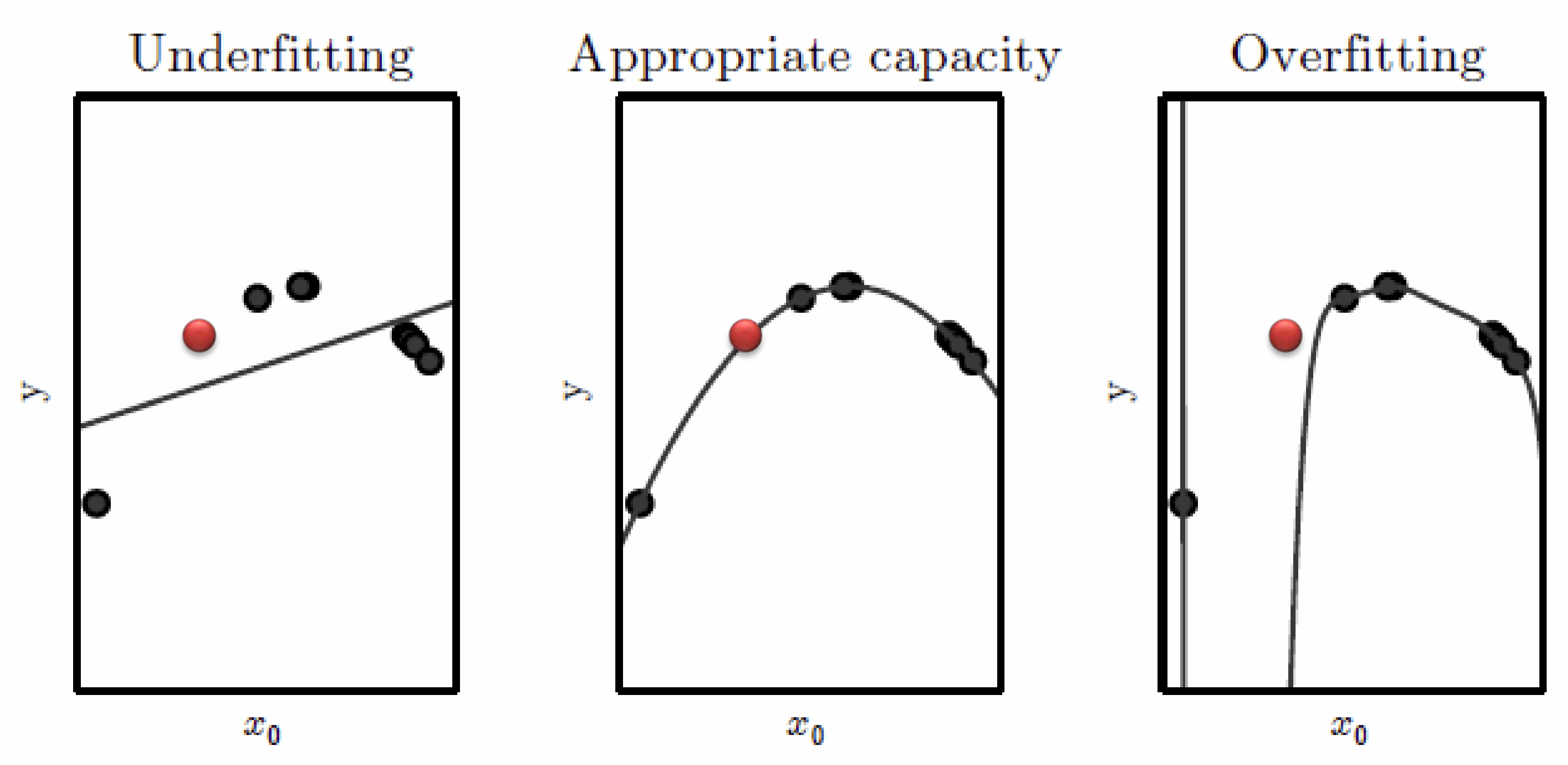
Generalization

- **In ML, generalization is the ability to perform well on previously *unobserved inputs***
  1. Making the training error small
  2. Making the gap between training and test error small

- ***Underfitting* occurs when the model is not able to obtain a sufficiently low error value on training set**

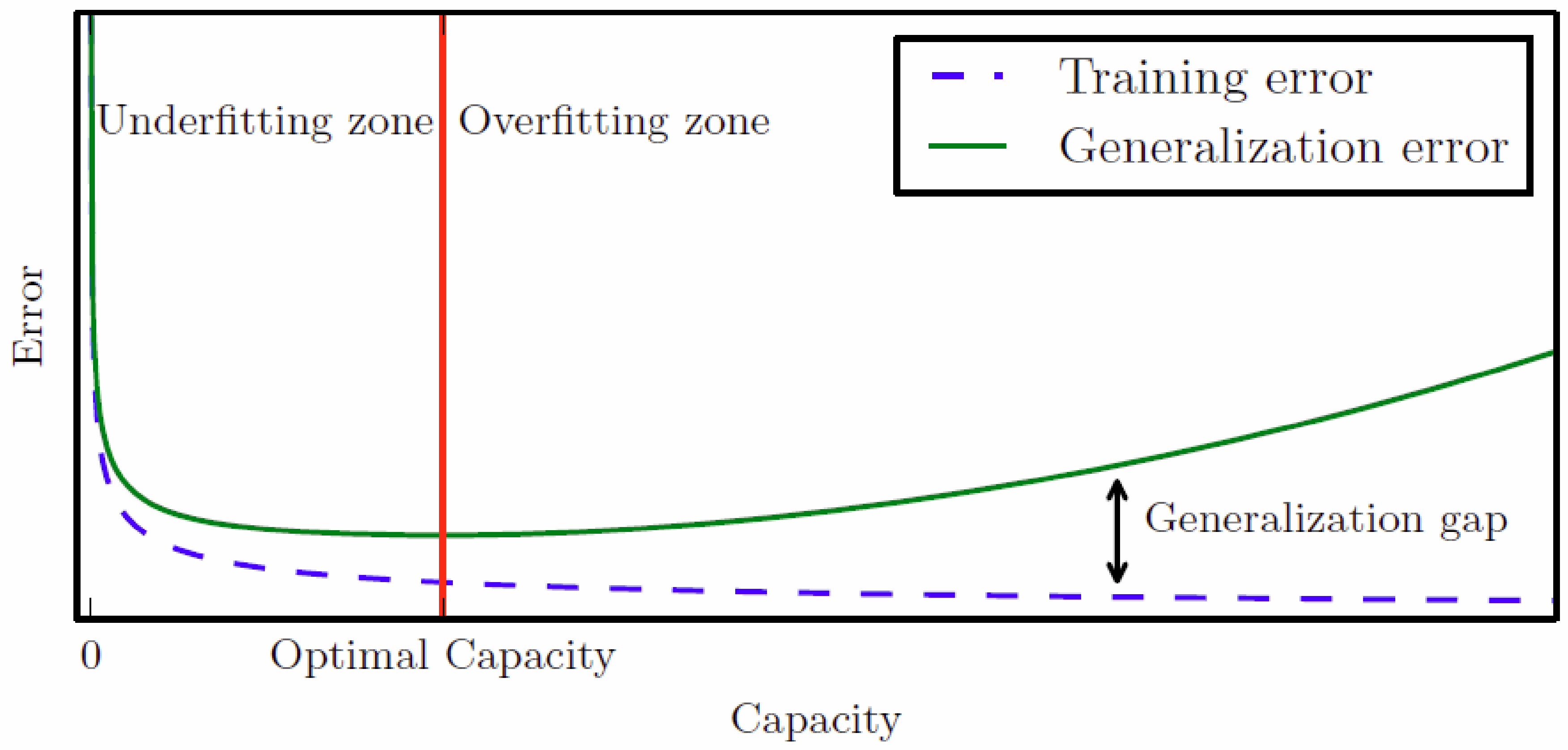
- ***Overfitting* occurs when the gap between the training error and the test error is too large**

Controlling a Model with its Capacity

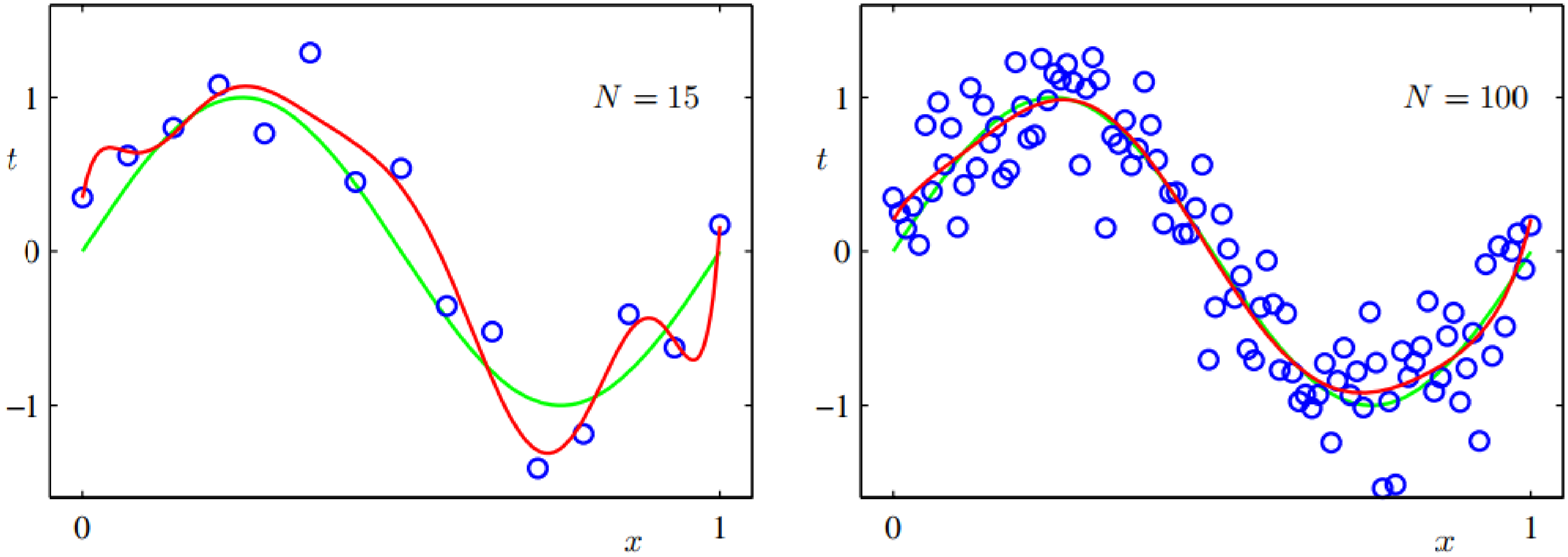


Linear (degree-1) Quadratic (degree-2) Polynomial (degree-9)

Relationship between Capacity and Error



Training Set Size and Generalization



Polynomial (degree-9)

Christopher, M. Bishop. *PATTERN RECOGNITION AND MACHINE LEARNING*. Springer-Verlag New York, Chapter 1.

Regularization

- **Regularization is any modification to prevent overfitting**

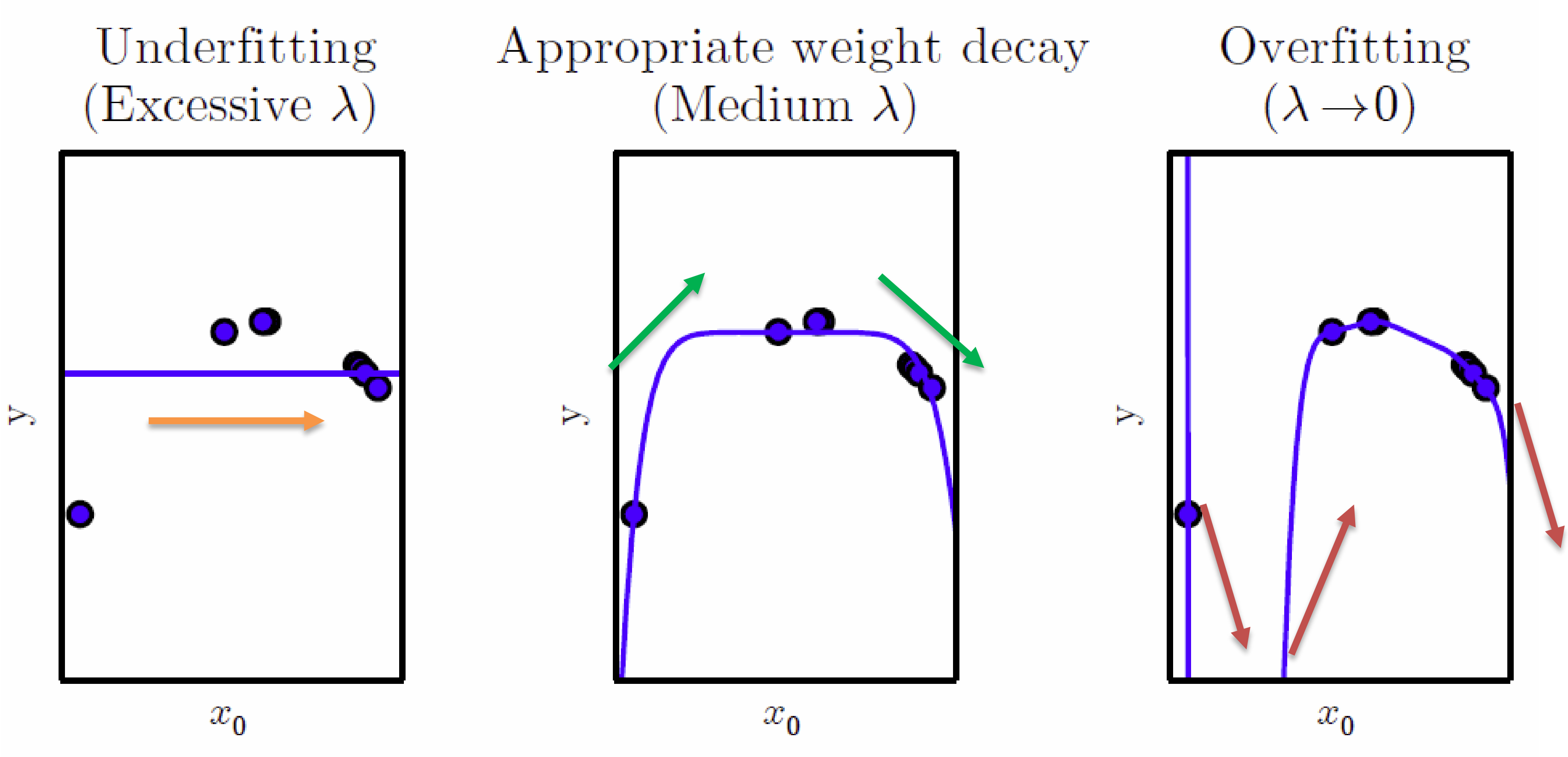
- **Regularization is able to *control the performance* of an algorithm**
  - Intended to reduce test error but not training error

- **Example: weight decay for regression problem**

$$J(\boldsymbol{w}) = MSE_{train} + \lambda \boldsymbol{w}^{\top} \boldsymbol{w}$$

  - $J(\cdot)$: cost function **to be minimized** on training
  - λ: control factor of the preference for smaller weight ($\lambda \geq 0$)
- Trades-off between fitting the training data and being small $\boldsymbol{w}$

9-degree Regression Model with Weight Decay



Large $\boldsymbol{w}$ over-fits to training data

Hyperparameters vs. parameters

- **Hyperparameters are higher-level properties for a model**
  - ➢ Decides model's capacity (or complexity)
  - ➢ Not be learned during training

  e.g., degree of regression model, weight decay

- **Parameters are properties of the training data**
  - ➢ Learned during training by a ML model

  e.g., weights of regression model

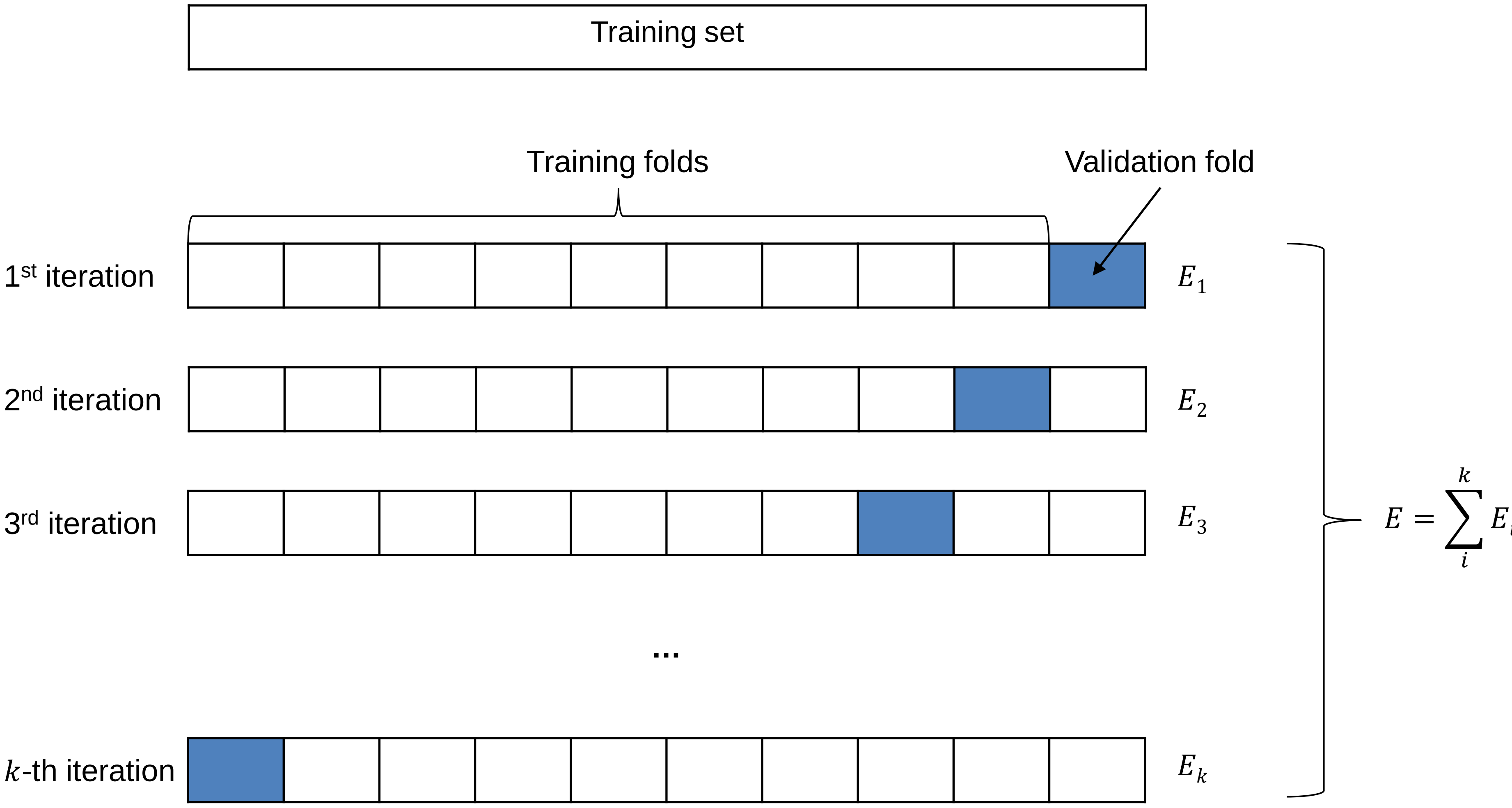
Setting Hyperparameters with Validation Set

- **Setting hyperparameters in training step is not appropriate**
  - Hyperparameters will be set to yield overfitting (e.g., higher degree of regression model, $\lambda \rightarrow 0$)

- **Test set will not be seen for training nor model choosing (hyperparameter setting)**

- **So, we need validation set that the training algorithm does not observe**
  1. Split validation set from training data
  2. Train a model with training data (not including validation set)
  3. Validate a model with validation set, update hyperparameters

k-fold Cross Validation



Estimation on Statistics

Point estimation

- Attempt to provide the singe best prediction of the true but unknown property of a model using sample data, $\{\boldsymbol{x}^1, \dots, \boldsymbol{x}^m\}$

Function estimation

- Types of estimation that predict the relationship between input and target variables

Point estimator

$$\widehat{\boldsymbol{\theta}}_{\boldsymbol{m}} = g(\boldsymbol{x}^1, \dots, \boldsymbol{x}^m)$$

$\widehat{\boldsymbol{\theta}}$: point estimator for the property of a model (e.g., expectation)

m: number of data elements

$\{\boldsymbol{x}^1, \dots, \boldsymbol{x}^m\}$: independent and identically distributed (i.i.d.) data points

$g(\cdot)$: any estimation function for the given data points

Properties of an Estimator

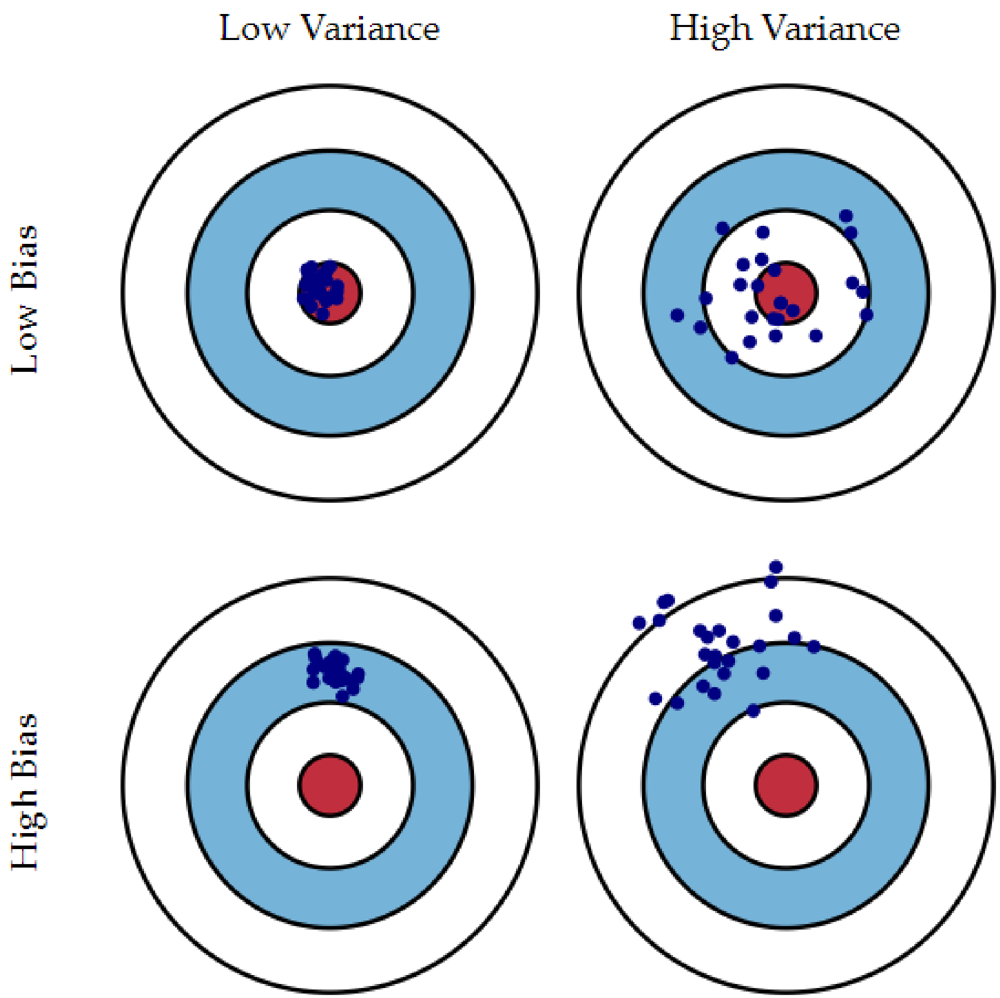
- **Bias** measures the expected deviation from **the true value** of θ

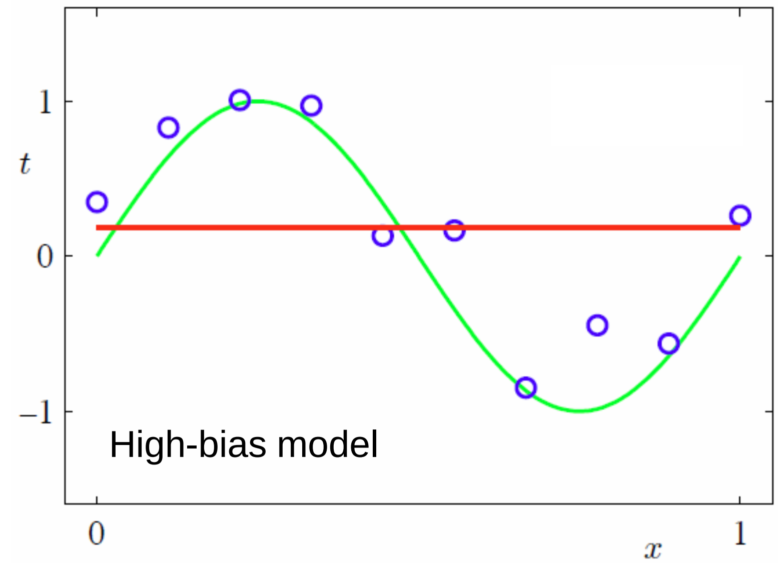
$$bias(\hat{\theta}_m) = \mathrm{E}[\hat{\theta}_m] - \theta$$

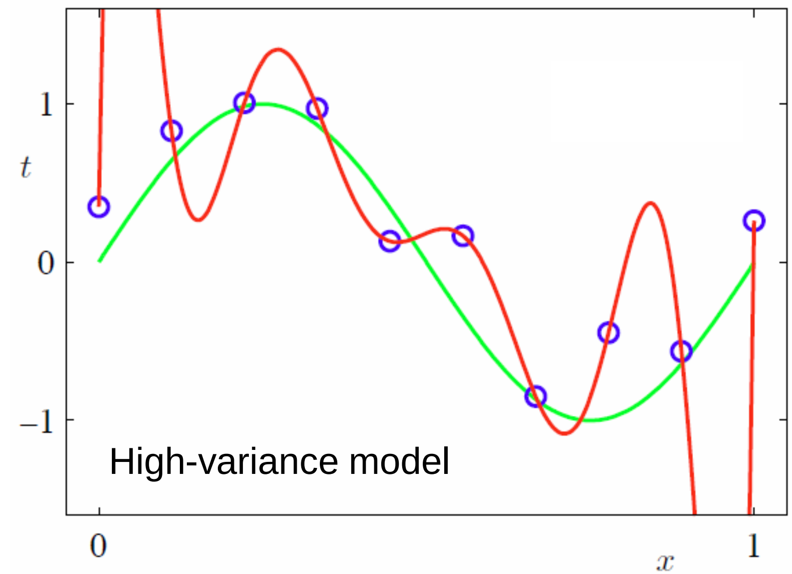
- **Variance** measures the deviation from the expected estimator value that **any particular sampling of the data** is likely to cause

$$Var(\hat{\theta})$$

Graphical Illustration of Bias and Variance

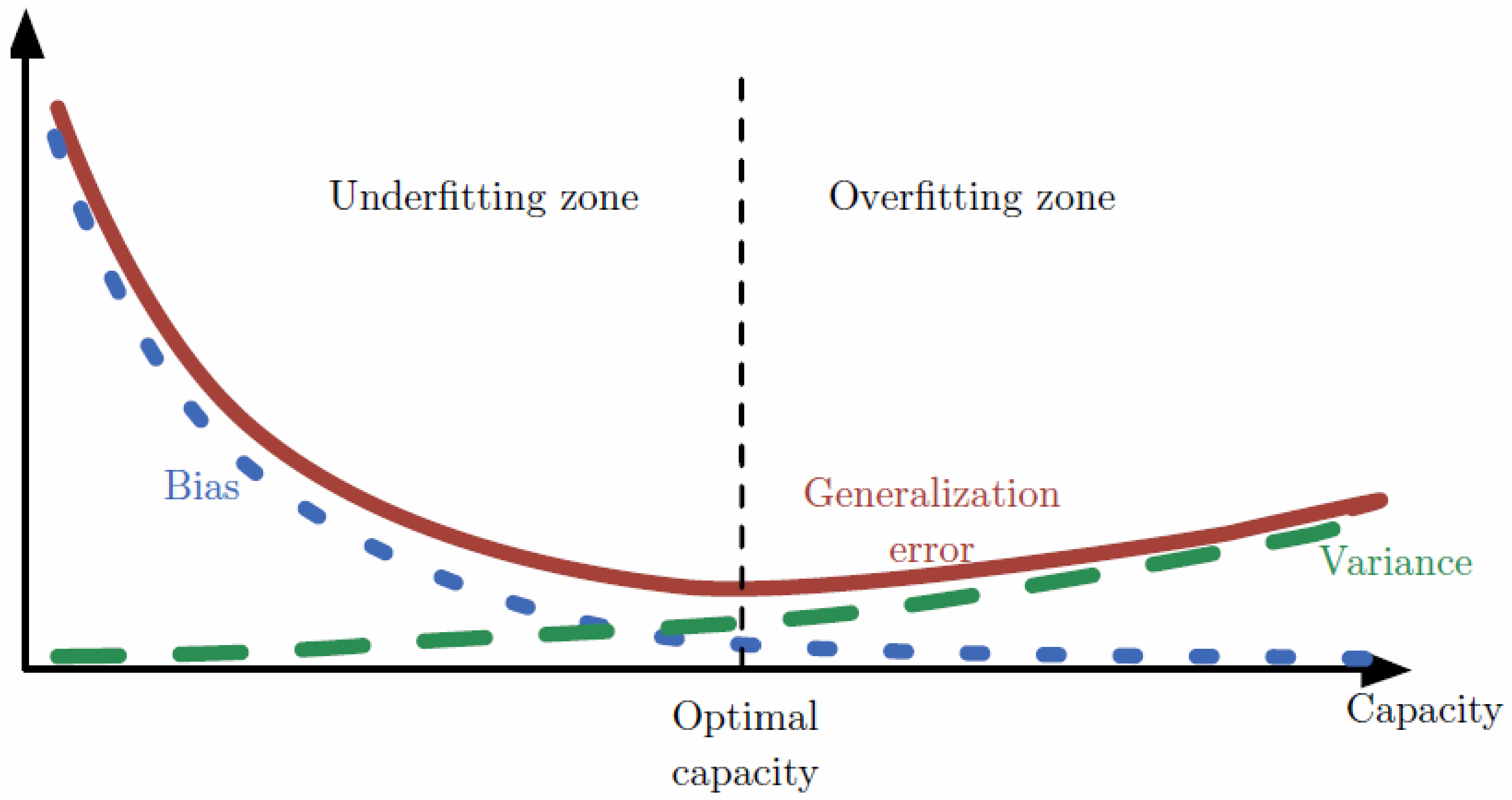






Scott Fortmann-Roe, Understanding the Bias-Variance Tradeoff, 2012

Bias-Variance Trade-off with Capacity



Bias & Variance on MSE

- **Let $\boldsymbol{h}(\cdot;\boldsymbol{w})$ be a regression model defined by $\boldsymbol{w}$**
  - $\boldsymbol{\theta} = t(\boldsymbol{x})$: the true (but unseen) distribution
  - $\widehat{\boldsymbol{\theta}} = E[h(\boldsymbol{x};\boldsymbol{w})]$: the estimation of $t(\boldsymbol{x})$

- **Then $\boldsymbol{MSE}$ of $\boldsymbol{g}$ is**

$$\boldsymbol{MSE} = \boldsymbol{E}\left[\left(\boldsymbol{h}(\boldsymbol{x};\boldsymbol{w}) - \boldsymbol{t}(\boldsymbol{x})\right)^2\right] = \boldsymbol{E}[(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta})^2]$$

- **Add and subtract $\boldsymbol{E}[\widehat{\boldsymbol{\theta}}]$ on the internal term**

$$\boldsymbol{E}[(\widehat{\boldsymbol{\theta}} - \boldsymbol{E}[\widehat{\boldsymbol{\theta}}] + \boldsymbol{E}[\widehat{\boldsymbol{\theta}}] - \boldsymbol{\theta})^2]$$

$$E\left[\left(\hat{\theta}-E[\hat{\theta}]+E[\hat{\theta}]-\theta\right)^2\right]$$

$$\begin{aligned}
= \quad & E[\ \hat{\theta}^2-\hat{\theta}E[\hat{\theta}]+\hat{\theta}E[\hat{\theta}]-\hat{\theta}\theta \\
& -\hat{\theta}E[\hat{\theta}]+E[\hat{\theta}]^2-E[\hat{\theta}]^2+E[\hat{\theta}]\theta \\
& +\hat{\theta}E[\hat{\theta}]-E[\hat{\theta}]^2+E[\hat{\theta}]^2-E[\hat{\theta}]\theta \\
& -\hat{\theta}\theta+E[\hat{\theta}]\theta-E[\hat{\theta}]\theta+\theta^2\ ]
\end{aligned}$$

$$\begin{aligned}
&= E\left[\left(\hat{\theta}-E[\hat{\theta}]\right)^2\right]+\left(E[\hat{\theta}]-\theta\right)^2 \\
&\qquad +2\left(E\left[\cancel{\hat{\theta}E[\hat{\theta}]}-\cancel{\hat{\theta}\theta}-\cancel{E[\hat{\theta}]^2}+\cancel{E[\hat{\theta}]\theta}\right]\right)
\end{aligned}$$

$$= \underbrace{E\left[\left(\hat{\theta}-E[\hat{\theta}]\right)^2\right]}_{Var(\hat{\theta})}+\underbrace{\left(E[\hat{\theta}]-\theta\right)^2}_{bias(\hat{\theta})^2}$$

Ways to Trade-off Bias & Variance

- **Cross-validation**
  - Highly successful on many real-world tasks

- **MSE of estimates**
  - MSE incorporates both bias and variance

$$
\begin{aligned}
MSE &= E\left[\left(\hat{\theta}_m - \theta\right)^2\right] \\
&= bias\left(\hat{\theta}_m\right)^2 + Var\left(\hat{\theta}_m\right)
\end{aligned}
$$

Maximum Likelihood Estimation

- **Maximum likelihood estimation (MLE) is a method of estimating the parameters of a statistical model**

$$\boldsymbol{w}_{ML} = \arg\max_{\boldsymbol{w}} \boldsymbol{p}_{\boldsymbol{model}}(\boldsymbol{X};\boldsymbol{w})$$

$$= \arg\max_{\boldsymbol{w}} \prod_{\boldsymbol{i}}^{\boldsymbol{m}} \boldsymbol{p}_{\boldsymbol{model}}(\boldsymbol{x}^{\boldsymbol{i}};\boldsymbol{w})$$

$$\equiv \arg\max_{\boldsymbol{w}} \sum_{\boldsymbol{i}}^{\boldsymbol{m}} \log\left(\boldsymbol{p}_{\boldsymbol{model}}(\boldsymbol{x}^{\boldsymbol{i}};\boldsymbol{w})\right)$$

- $p_{data}(\boldsymbol{x})$: true but unknown data-generating distribution
- $\boldsymbol{X} = \{\boldsymbol{x}^{\boldsymbol{1}}, \dots \boldsymbol{x}^{\boldsymbol{m}}\}$: drawn independently from $p_{data}(\boldsymbol{x})$
- $p_{model}(\boldsymbol{x};\boldsymbol{w})$: a probability distribution estimating $p_{data}(\boldsymbol{x})$

Linear Regression as Maximum Likelihood

- **Instead of single prediction $\hat{y}$, we consider of the model as producing a conditional distribution, $\boldsymbol{p}(y|\boldsymbol{x})$**

- **Let $\boldsymbol{p}(y|\boldsymbol{x}) = \mathcal{N}(y \mid \hat{y}(\boldsymbol{x};\boldsymbol{w}), \sigma^2)$**
  - $\hat{y}(\boldsymbol{x};\boldsymbol{w})$: the prediction of the mean of the Gaussian
  - σ^2: variance of $\mathcal{N}$ chosen by user

- **Log-likelihood:**

$$\sum_i^m \log(\boldsymbol{p}_{\boldsymbol{model}}(\boldsymbol{x}^i;\boldsymbol{w})) = \sum_i^m \log(y^i|\boldsymbol{x}^i;\boldsymbol{w})$$

$$= -m\log(\sigma) - \frac{m}{2}\log 2\pi - \sum_i^m \frac{\|\hat{y}^i - y^i\|^2}{2\sigma^2}$$

- **Cf. $\boldsymbol{MSE}$:**

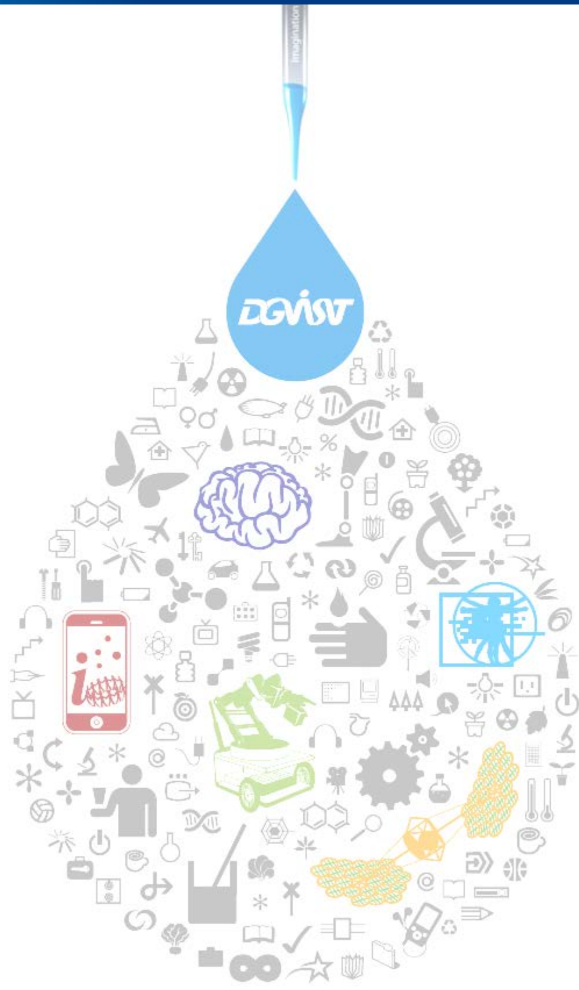
$$\boldsymbol{MSE}_{\mathbf{train}} = \frac{1}{m}\sum_i \|\hat{y}^i - y^i\|^2$$

The same estimate of the parameter w

Contents on Part 2

- Bayesian statistics
- Supervised learning algorithms
- Unsupervised learning algorithms
- Stochastic gradient descent
- Building a machine learning algorithm
- Challenges motivating deep learning

InfoLab



Q&A

Thank you

대구경북과학기술원
Daegu Gyeongbuk
Institute of Science & Technology