Chap.2 모델의 검증과 평가 (Model Test and Evaluation)

방 수 식 교수

(bang@tukorea.ac.kr)

한국공학대학교 전자공학부

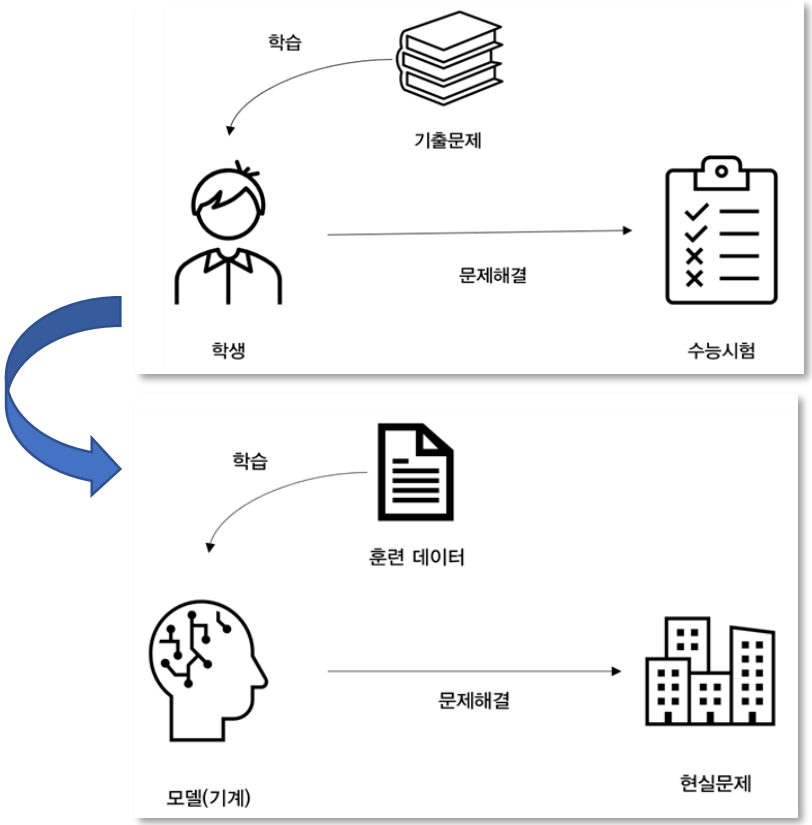
2024년도 1학기

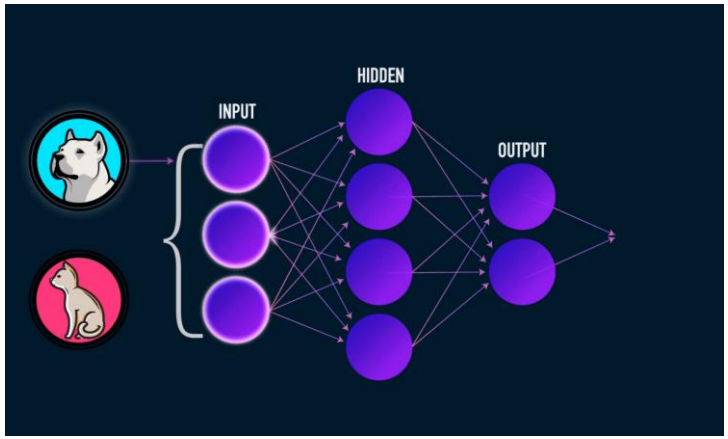
머신러닝실습 & 인공지능설계실습1

머신러닝의 목표와 모델

- 머신러닝 모델
  - ➢ 입력(Input)에 따라 출력(Output)을 생산 => 전달함수

- 머신러닝의 목표
  - ➢ 입력에 따른 출력이 실측값과 동일한 결과를 만드는 전달함수(모델) 도출





머신러닝 모델의 오차와 MSE

- **오차(Error)**
  - 전체 데이터에 대해, 실측값 – 모델의 예측값
  - 다양한 측정 방법이 가능하며 일반적으로 [0,1] 사이의 값을 갖도록 설계

- **평균 제곱 오차 (Mean squared error, MSE)**
  - 머신러닝 모델의 대표적인 성능 평가 기준
  - 데이터 세트 $D = \{(\boldsymbol{x}_0, y_0), (\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_{N-1}, y_{N-1})\}$
  - 모델 f의 이상적인 성능은 $f(\boldsymbol{x}) = y$, 즉 예측값이 실제값과 동일,
    - 실제로는 오차 발생
  - 정의

$$\epsilon_{MSE}(f; D) = \frac{1}{N}\sum_{n=0}^{N-1}(f(\boldsymbol{x}_n) - y_n)^2$$

Training set과 Test set

- Training set
  - 모델을 **학습(Training)**하기 위한 데이터 set
    => Weight의 Update

$$w_0[t+1] = w_0[t] - \alpha \frac{\partial}{\partial w_0} \epsilon_{MSE}(w_0, w_1)$$

(New: $w_0[t+1]$, Old: $w_0[t]$, Learning Rate: α, Old: w_0, w_1)

- Test set
  - 모델을 **평가(Test)**하기 위한 데이터 set => **결정된 Weight로 모델 평가**

Classification(분류) 모델에서의 평가

○ 오차율

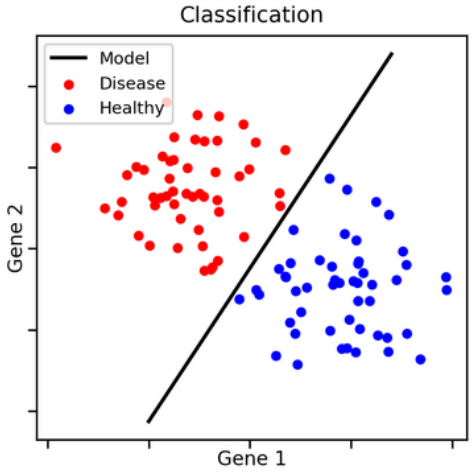
- 모든 데이터 중 잘못 분류한 데이터의 비율

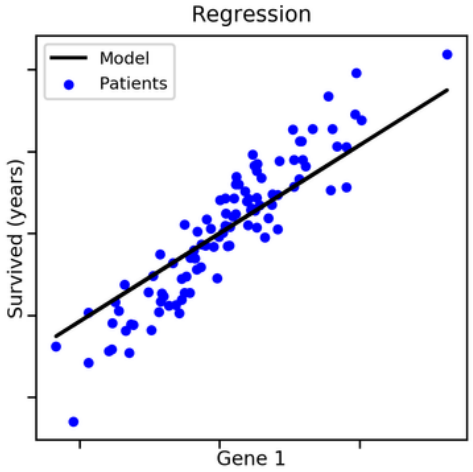
$$P_{err}(f;D) = \frac{1}{N}\sum_{n=0}^{N-1} \boldsymbol{I}(f(x_n) \neq y_n)$$

○ 정확도

- 모든 데이터 중 성공적으로 분류한 데이터의 비율

$$P_{acc}(f;D) = \frac{1}{N}\sum_{n=0}^{N-1} \boldsymbol{I}(f(x_n) = y_n) \; = 1 - P_{err}(f;D)$$





*분류는 데이터를 서로 **구별**하는 것*

*회귀는 데이터에 대한 경향을 **예측**하는 것*

Precision(정밀도) & Recall(재현율)

- 예시) 머신러닝 기반 코로나 검출 모델

		모델의 예측 결과	
		양성 (Positive)	음성 (Negative)
데이터의 실제 결과	양성 (Positive)	진짜 양성 (TP, True Positive)	가짜 음성 (FN, False Negative)
	음성 (Negative)	가짜 양성 (FP, False Positive)	진짜 음성 (TN, True Negative)

전체 데이터의 수 N = TP+FP+TN+FN

- 정확도 = $\frac{TP+TN}{N}$
- 오차율 = $1 - \frac{TP+TN}{N} = \frac{FP+FN}{N}$

- TP(True Positive): 실제 양성을 머신러닝 모델이 양성으로 판별
- FP(False Positive): 실제 음성을 머신러닝 모델이 양성으로 판별
- FN(False Negative): 실제 양성을 머신러닝 모델이 음성으로 판별
- TN(True Negative): 실제 음성을 머신러닝 모델이 음성으로 판별

Precision(정밀도) & Recall(재현율)

- $Precision = \frac{TP}{TP+FP}$
  - ➢ 모델이 양성으로 분류한 것 중 실제 양성 비율
- $Recall = \frac{TP}{TP+FN}$
  - ➢ 실제 양성 중 모델의 양성 판별 비율

		모델의 예측 결과	
		양성 (Positive)	음성 (Negative)
데이터의 실제 결과	양성 (Positive)	진짜 양성 (TP, True Positive)	가짜 음성 (FN, False Negative)
	음성 (Negative)	가짜 양성 (FP, False Positive)	진짜 음성 (TN, True Negative)

두 지표가 둘다 높은 모델이 일반적으로 좋은 모델 (항상은 아님)

- 예시) 코로나 데이터 100 건에 대해서 실제 양성이 5명, 음성이 95명일 때,

		모델의 예측 결과	
		양성 (Positive)	음성 (Negative)
데이터의 실제 결과	양성 (Positive)	0	5
	음성 (Negative)	1	94

- Accuracy(정확도): 94 %
- Precision: 0 %
- Recall: 0 %

Precision(정밀도) & Recall(재현율)

- 예시) 코로나 데이터 100 건에 대해서 실제 양성이 95명, 음성이 5명일 때,

1차적 문제: Data Unbalance(데이터 불균형)

		모델의 예측 결과	
		양성 (Positive)	음성 (Negative)
데이터의 실제 결과	양성 (Positive)	95	0
	음성 (Negative)	5	0

- Accuracy: 95 %
- Precision: 95 %
- Recall: 100 %

???

- Specificity(특이도) = $\frac{TN}{FP+TN}$ => 위의 예제에서 Specificity: 0 %
- FPR(False positive rate) = 1 – Specificity = 1 - $\frac{TN}{FP+TN}$ = $\frac{FP}{FP+TN}$

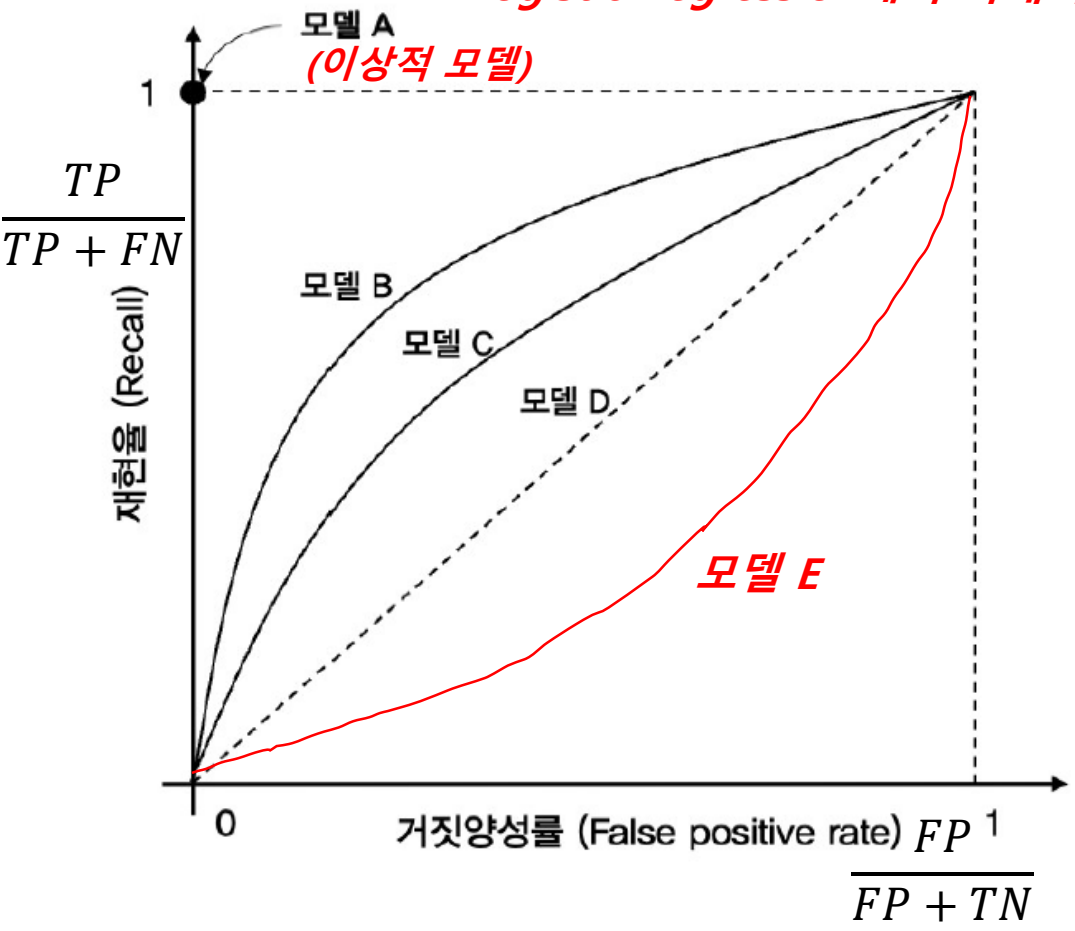
Recall이 높으면서 FPR이 낮은 모델 => Good 모델

ROC Curve

- Recall이 높으면서 FPR이 낮은 모델을 찾는 법
  - ➢ FPR vs Recall 그래프, x축, x=1의 면적이 넓을 수록 좋다.

분류 기준 값(Threshold)을 변경하면서 Curve를 plot
=> Logistic Regression에서 자세히 설명

		모델의 예측 결과	
		양성 (Positive)	음성 (Negative)
데이터의 실제 결과	양성 (Positive)	진짜 양성 (TP, True Positive)	가짜 음성 (FN, False Negative)
	음성 (Negative)	가짜 양성 (FP, False Positive)	진짜 음성 (TN, True Negative)

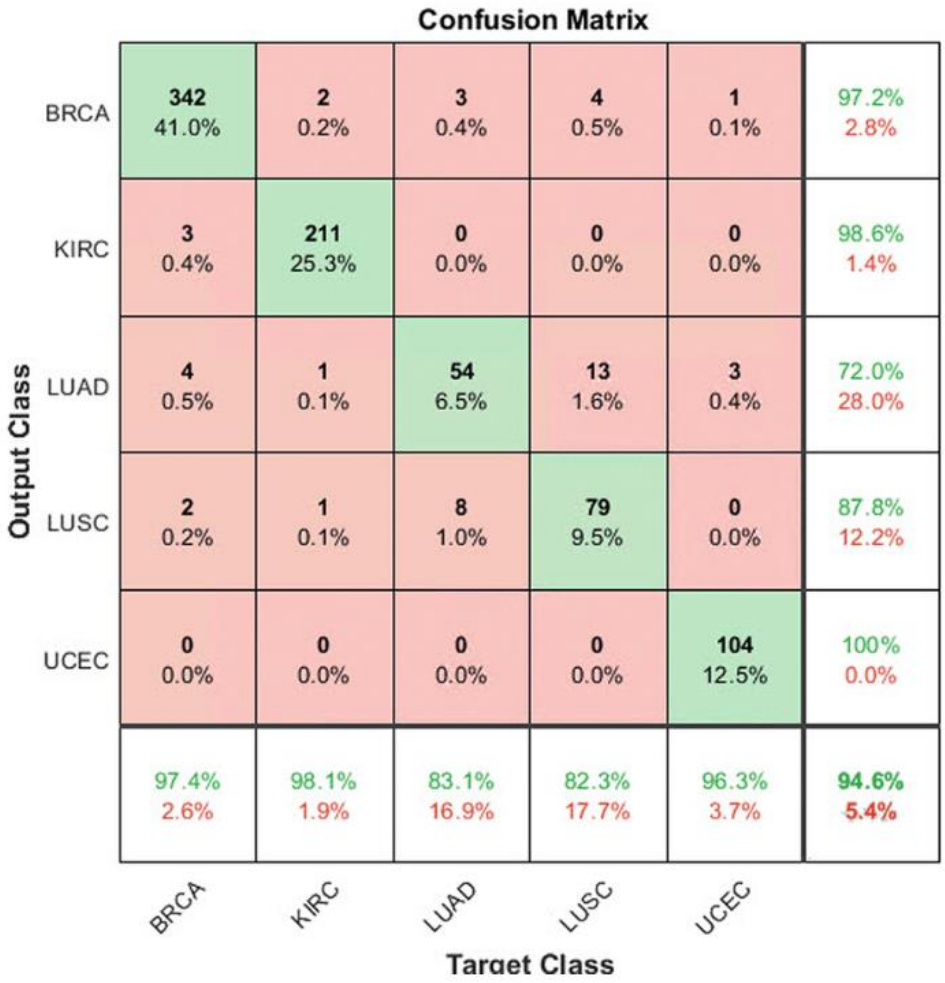


1) 모델 D는 어떤 모델을 의미하는 건가?

2) 모델 E는 존재할 수 없나?

Multi-Class Classification

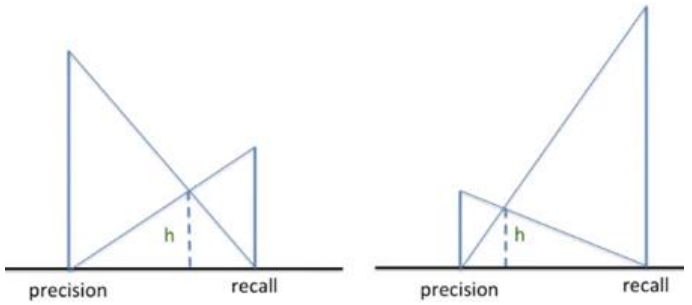
- **Confusion Matrix (혼동행렬)**



F1 Score

- 다중 분류(Multi-class Classification) 문제에서의 모델 평가
  - Precision과 Recall의 조화평균
  - $F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$

전체 class에 대한 precision과 recall은 각 class에 대한 평균값



둘 다 높아야 h(조화평균)가 크다

- $Precision = \frac{TP}{TP+FP}$, $Recall = \frac{TP}{TP+FN}$

A class에 대한 TP = 9

actual class (input) \ predictions (output)	A	B	C	D
A	9	1	0	0
B	1	15	3	1
C	5	0	24	1
D	0	4	1	15

A class에 대한 FP = 6

actual class (input) \ predictions (output)	A	B	C	D
A	9	1	0	0
B	1	15	3	1
C	5	0	24	1
D	0	4	1	15

A class에 대한 FN = 1

A class에 대한 Precision = 9/15
A class에 대한, Recall = 1/10

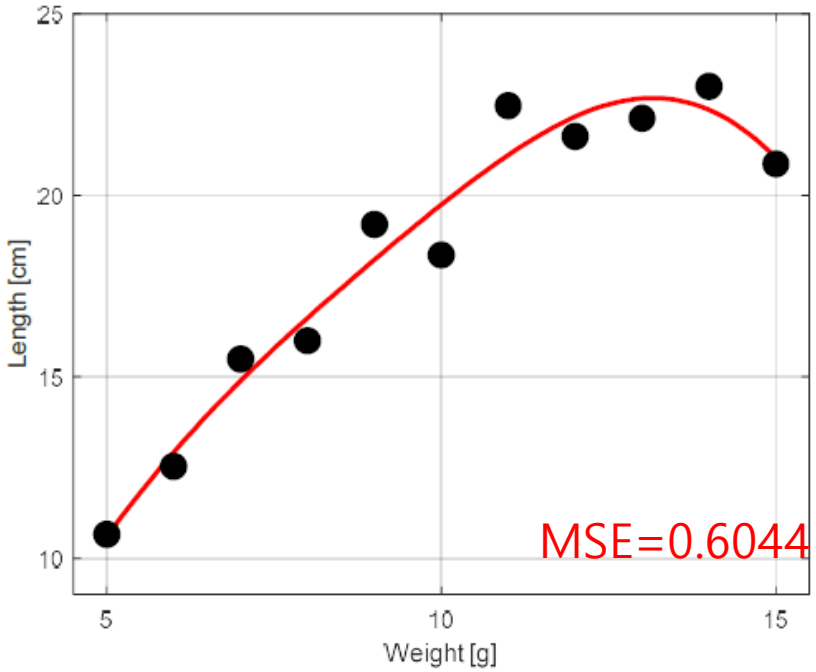
[복습] 기저함수 수(K)에 따른 결과 비교

$$\hat{y} = 27.02\, e^{-\frac{1}{2}\left(\frac{x-5}{5}\right)^2} + 3.46\, e^{-\frac{1}{2}\left(\frac{x-10}{5}\right)^2} + 39.08\, e^{-\frac{1}{2}\left(\frac{x-15}{5}\right)^2} - 23.82$$

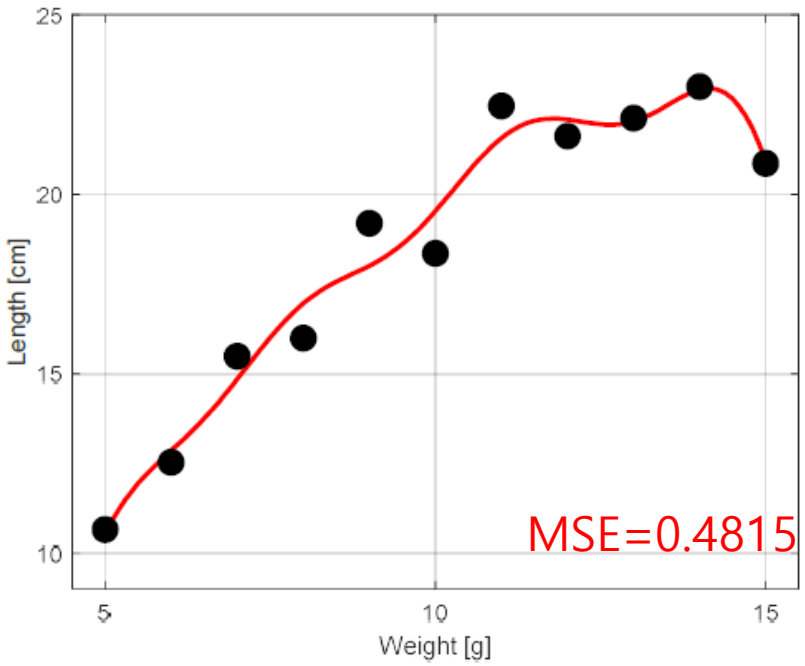
$\phi_0(x)$ $\phi_1(x)$ $\phi_2(x)$

$$w = \begin{bmatrix} 27.02 \\ 3.46 \\ 39.08 \\ -23.82 \end{bmatrix}$$

$K = 3$



$K = 8$



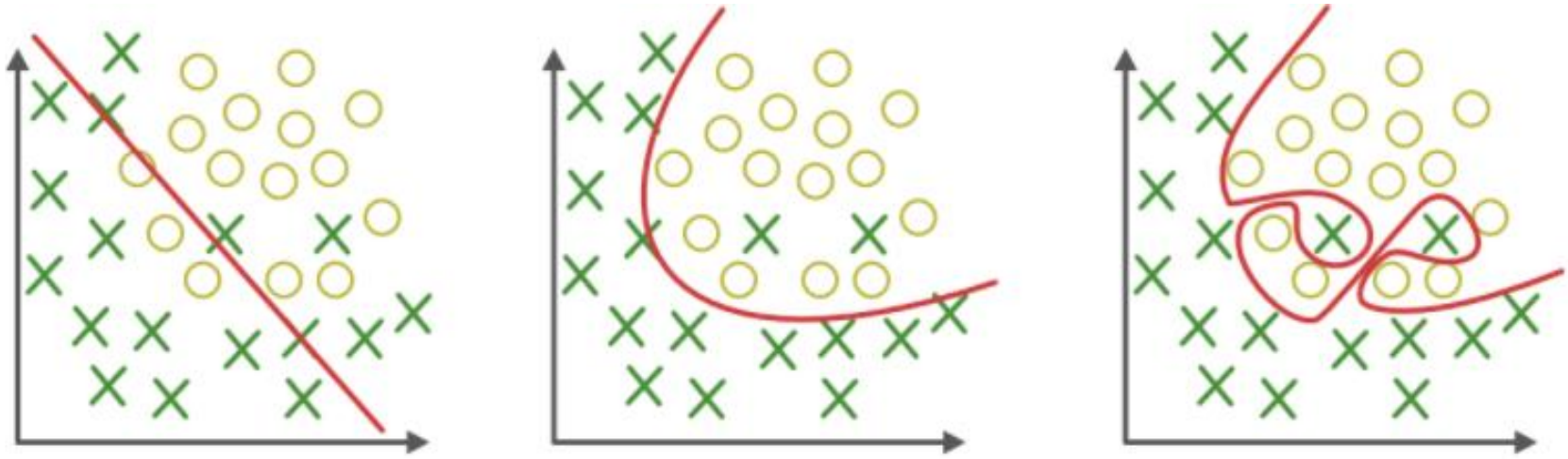
MSE가 작다고 무조건 좋은 것일까?

머신러닝 모델에서의 정확도 문제

- **훈련 오차 vs. 일반화 오차**
  - 일반화 오차를 줄이는 것이 궁극적 목표임
  - 그러나, 훈련 과정에서는 훈련 데이터만 경험할 수 있음
    - 실전 데이터를 경험할 수 없음
  - 훈련 데이터의 학습을 통해 일반화 성능을 최대화해야 함
    - 머신러닝의 근본적 문제의 원인
  - 두 가지 중 하나의 문제에 빠질 위험이 있음
    - 학습 능력이 부족하거나 => **과소적합의 문제 (underfitting)**
    - 학습이 과하거나... => **과적합의 문제 (overfitting)**

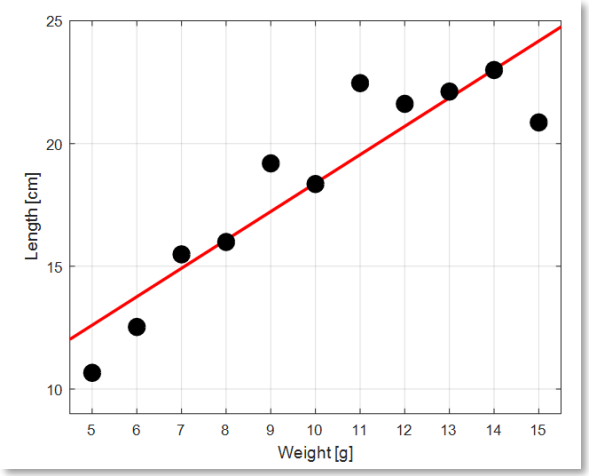
Underfitting vs Overfitting

- 다음 중 바람직한(일반화된) fitting line은?



Underfitting (과소적합)

- 학습능력이 부족한 경우
  - 추의 무게와 용수철의 관계
    - 최적 매개변수를 찾았지만, 모델 자체의 학습 능력이 부족함



– 이 모델에 따르면, 추의 무게가 무한하면 길이도 무한해짐

Underfitting이 일어난 원인은 무엇인가?

  - 이미지 분류 문제

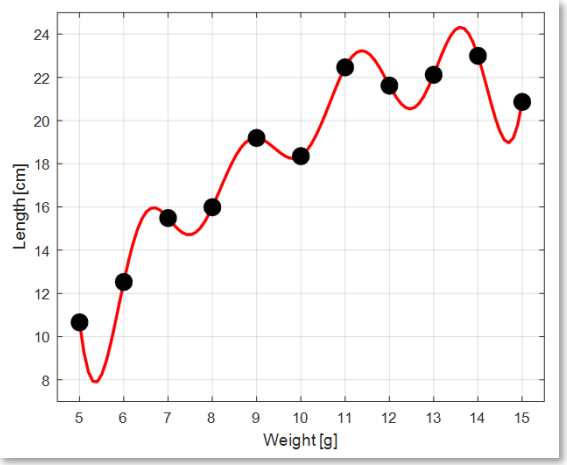
훈련 데이터

새로운 데이터

학습기 예측 결과
→ 이것은 나뭇잎이다.
이유
→ 초록색 둥근 모양은 나뭇잎이다.

Overfitting (과대적합)

- 학습이 과도한 경우
  - 추의 무게와 용수철의 관계
    - 훈련 데이터를 완벽하게 학습한 모델



– 이 모델은 훈련 데이터만 정확하게 맞춤
– 훈련 오차는 최소이나, 일반화 오차는 커짐

Overfitting이 일어난 원인은 무엇인가?

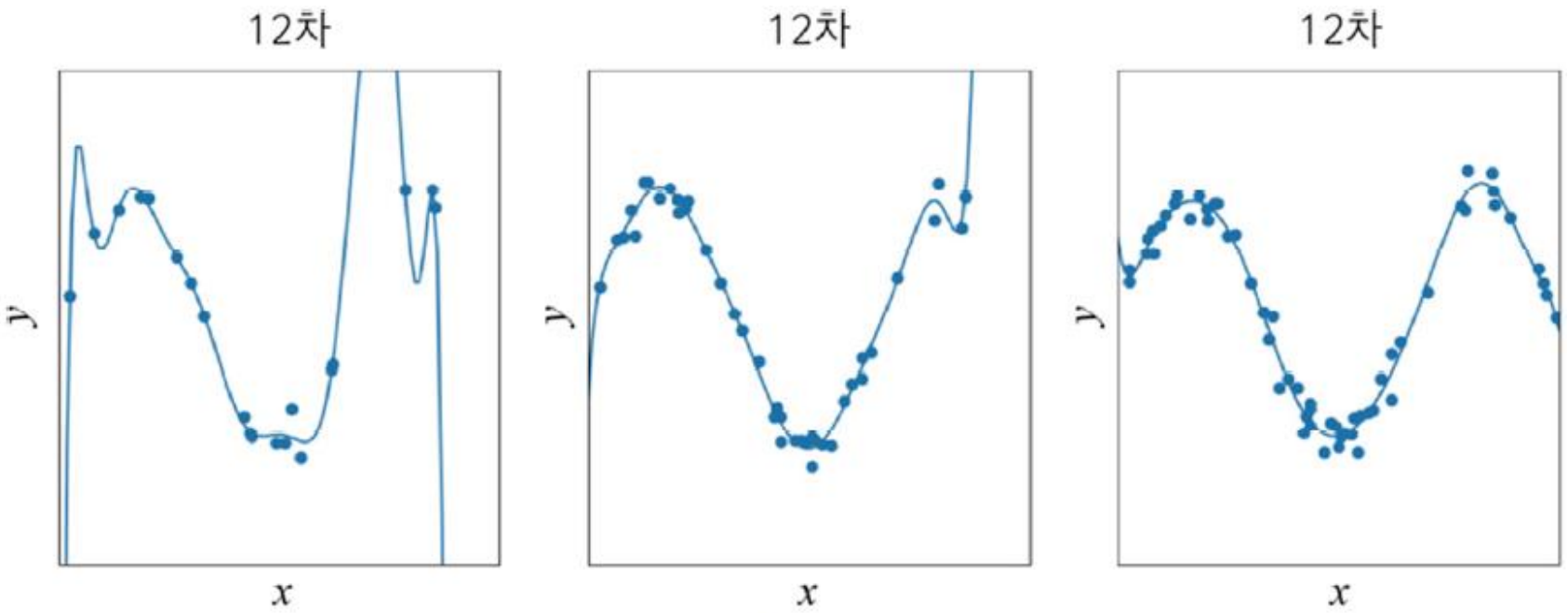
  - 이미지 분류 문제

훈련 데이터 새로운 데이터

학습기 예측 결과
→ 이것은 나뭇잎이 아니다.
이유
→ 나뭇잎의 테두리는 톱니 모양이다.

Training 데이터 증가

- Training을 위한 데이터를 늘리면 일반화에 유리

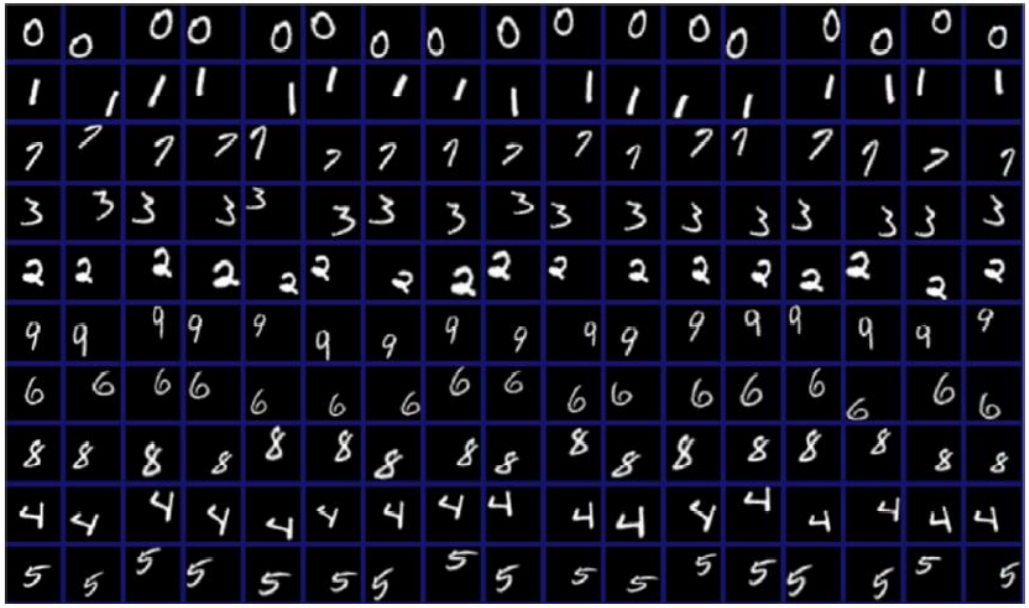


(a) 훈련집합의 크기=20 (b) 훈련집합의 크기=40 (c) 훈련집합의 크기=60

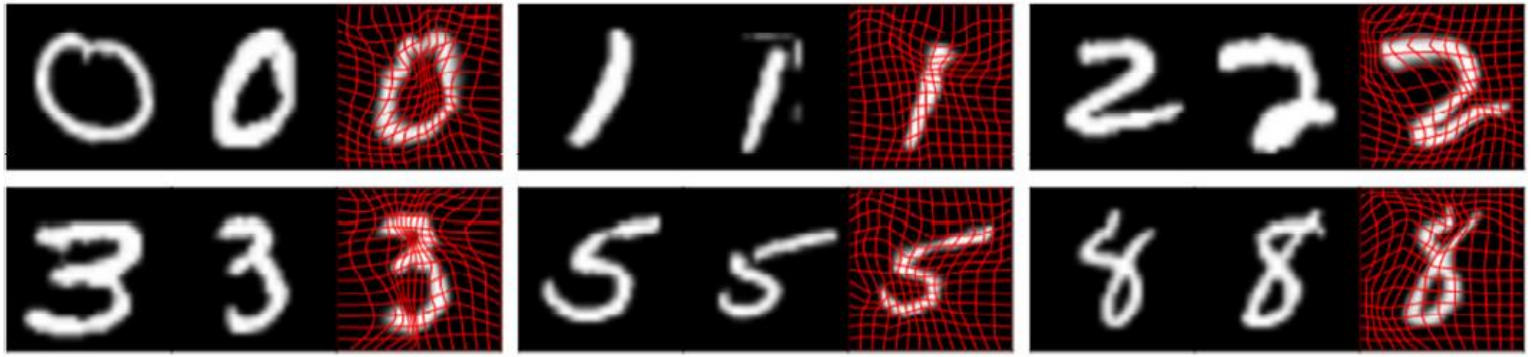
데이터를 더 이상 수집하기 힘든 경우?

Data Augmentation

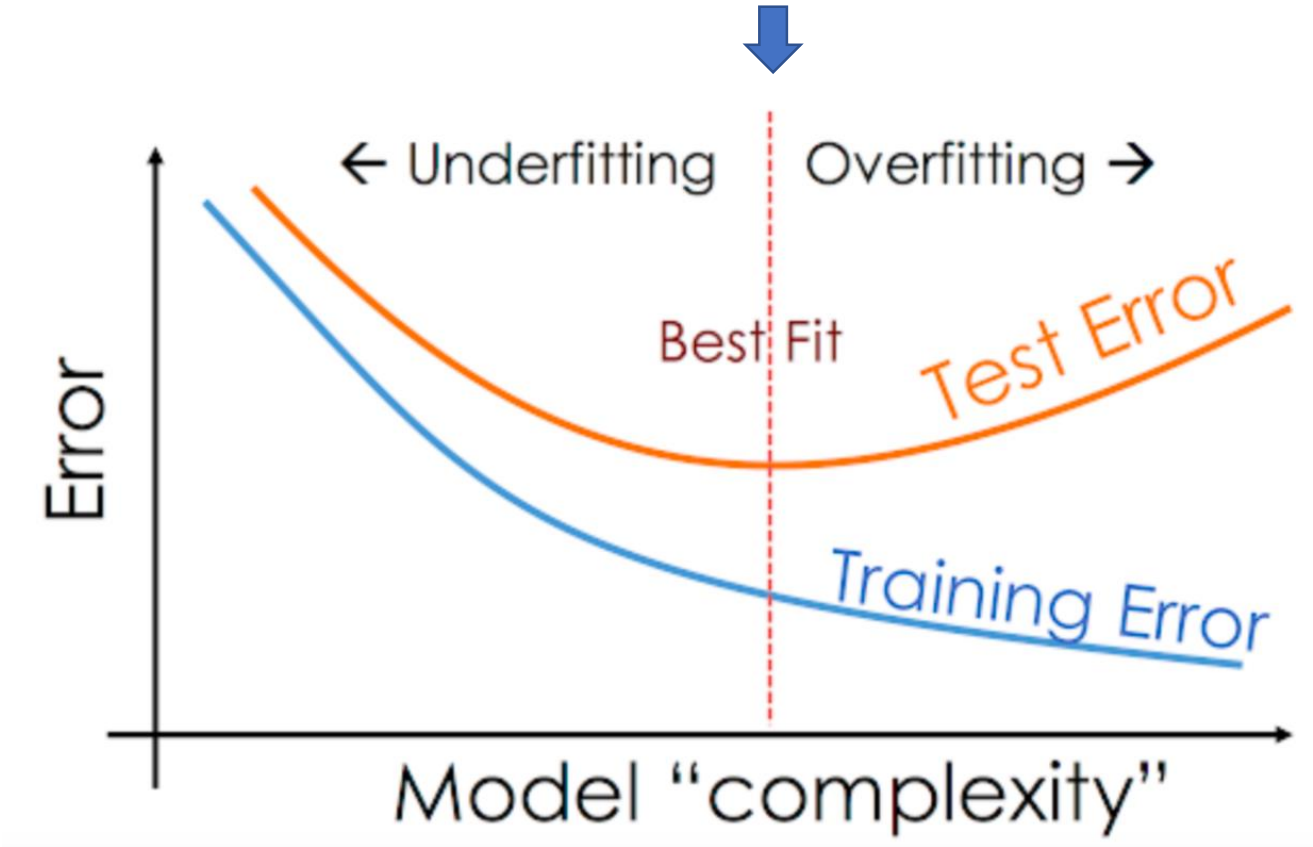
- MNIST에 Affine Transformation(이동, 회전, 크기)을 적용



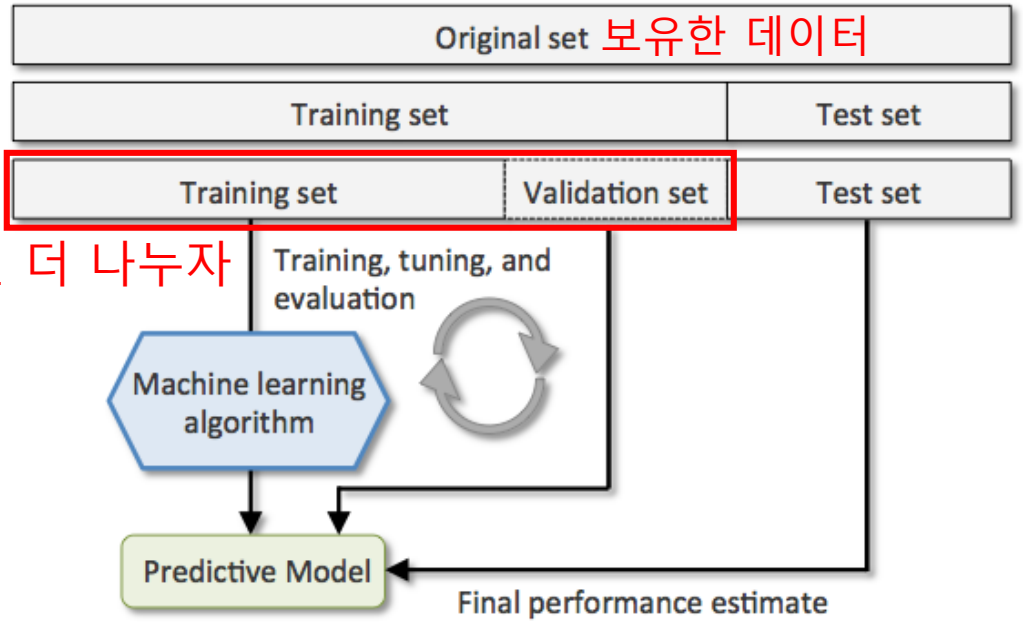
- 비선형 변환

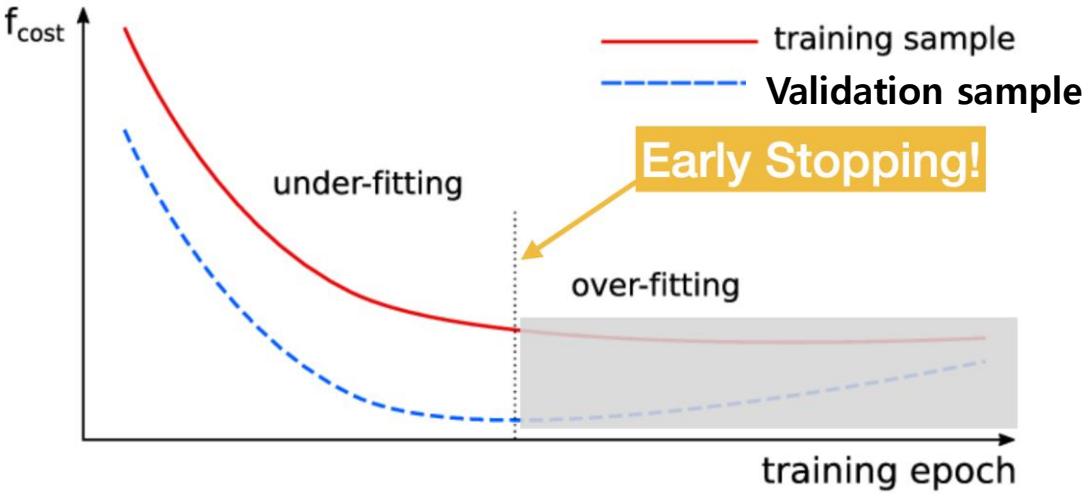


모델을 어떻게 검증하며 학습할 것인가?



Hold-out Validation





Training set vs Validation set vs Test set

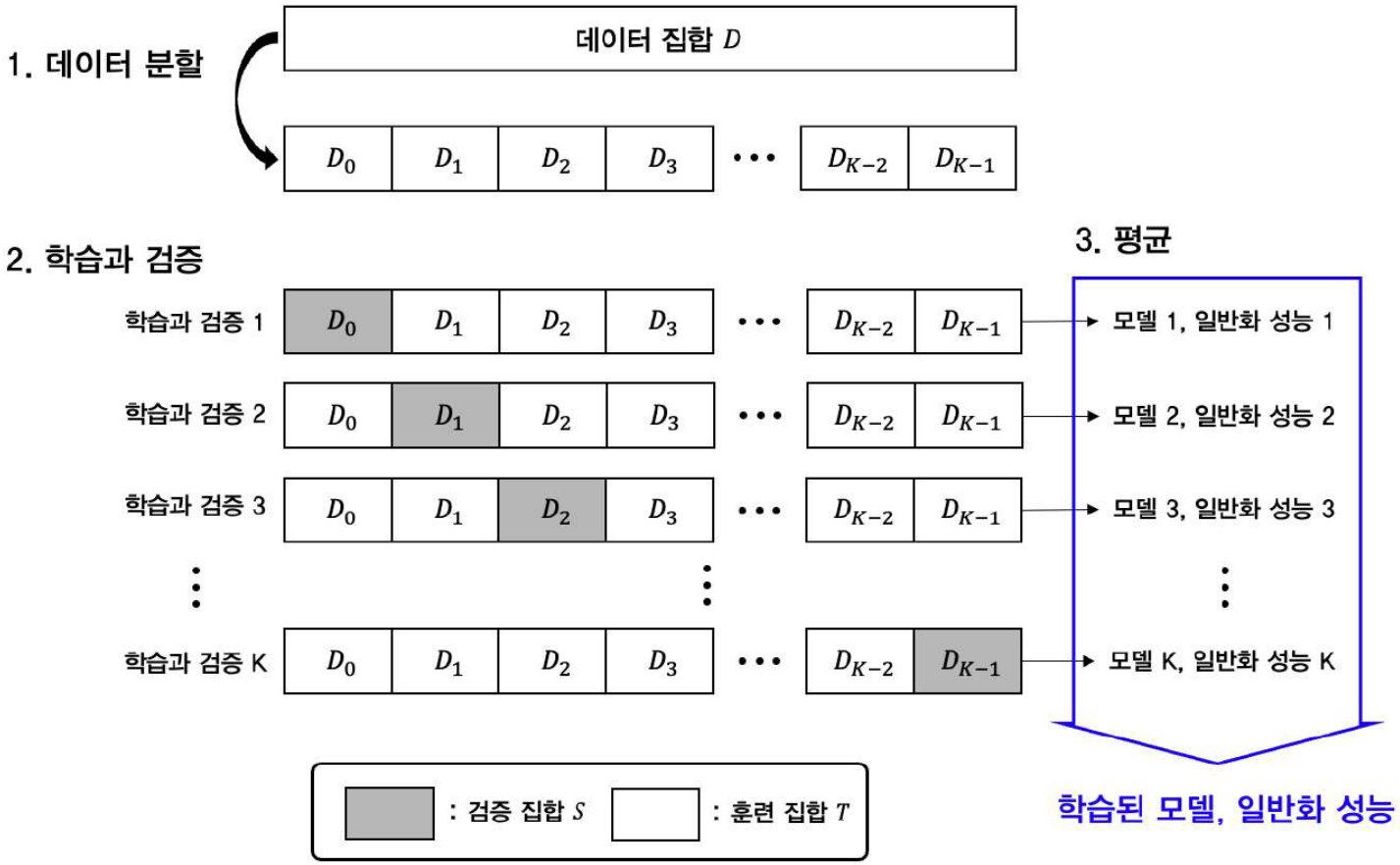
- Original Set
  - 머신러닝에 활용되는 전체 데이터 집합
- Training Set
  - 머신러닝 모델 학습(Weight Update)에 사용되는 데이터 집합
- Validation Set
  - 모델 학습 도중 모델을 **중간평가(검증)** 하기위한 데이터 집합
  - **Weight update에는 활용되지 않음**
- Test Set
  - 최종적으로 학습된 모델의 성능을 평가하기 위한 데이터 집합

각 데이터 set 간에 공통된 데이터는 존재하지 않아야 함

K-fold Cross Validation

- 장점: 모델 학습을 위한 데이터 수를 확보할 수 있다.
- 단점: k값에 따라 연산량이 많아진다.

Validation set으로 나누지 않아도 된다.

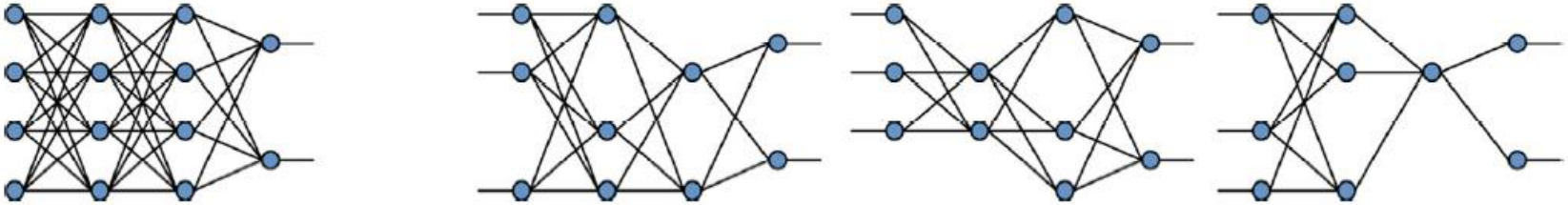


❖ **Leave one out cross validation(LOOCV): K와 데이터 수가 같을 경우를 지칭**

그 외 방법 (Dropout)

Dropout 기법

- 입력층과 은닉층의 노드 중 일정 비율을 임의로 선택하여 제거
- 남은 부분 신경망을 학습
- 많은 부분 신경망을 만들고, 예측 단계에서 앙상블 결합하는 기법으로 볼 수 있음



(a) 원래 신경망(4-4-4-2 구조) (b) 드롭아웃된 3개의 신경망 예시

신경망에서 다시 설명할 예정

Chap.2 실습

- Chap.1 실습에서의 데이터 set "lin_regression_data_01.csv" 에 대해서

1) lin_regression_data_01.csv 데이터에 대해 Random noise를 추가하여 데이터 수를 20배 (50개 -> 1000개) 증강하고, Original Set과 Augmented(증강) set을 하나의 그래프에 나타내라. (Noise 크기에 따른 데이터 set 변화 분석 필수)

2) 사용자 지정함수를 활용하여, 보유한 Data set을 *사용자의 입력비율* 에 따라 Training set, Validation set, Test set로 분할해주는 함수를 구현하고, 5:3:2로 분할된 데이터를 하나의 그래프에 나타내라. (예시: 100개의 데이터에 대해 *6,2,2*를 입력하면 60, 20, 20개의 데이터 set으로 분할, 이 때 분할되는 데이터는 실행마다 **랜덤하게 분배** 되어야함.)

3) lin_regression_data_01.csv 데이터에 대해서 8:0:2로 set을 나누고, Chap.1에서 구현한 가우시안 기저함수 모델 코드를 응용하여, 본 ppt 자료 18page 같은 그래프를 그리고, 최적의 K(가우시안 기저함수 개수) 를 도출하라.

각 실습문제에 대해 code(+주석), 그래프, 분석 내용 등은 필수적으로 포함시킬 것