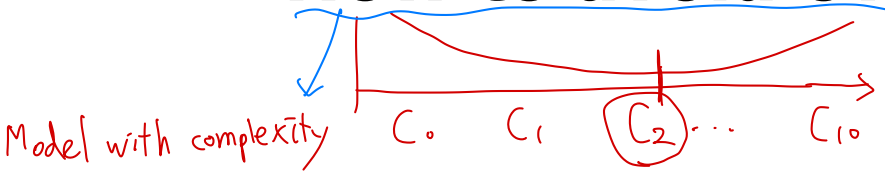
Method

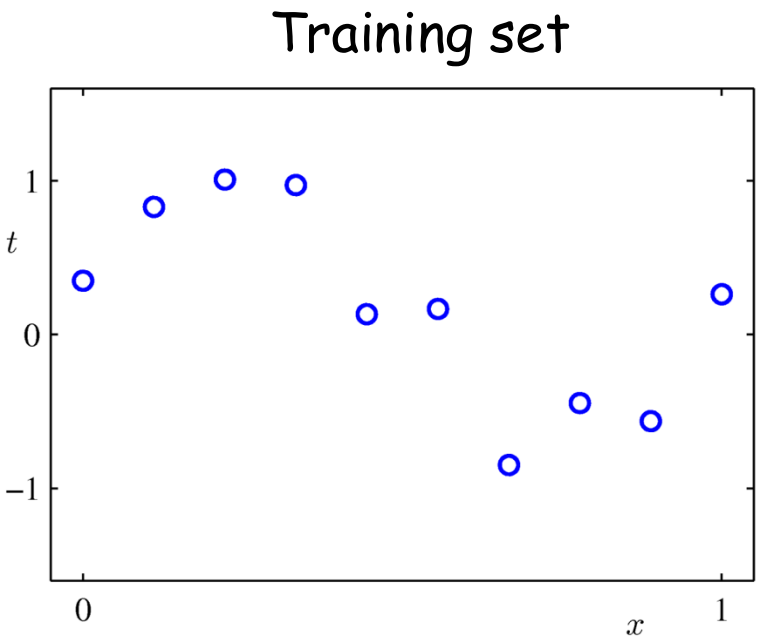
Regularization
- How to avoid overfitting -



→ 문제점 : 모든 모델을 빌드해야된다. (cost ↑)

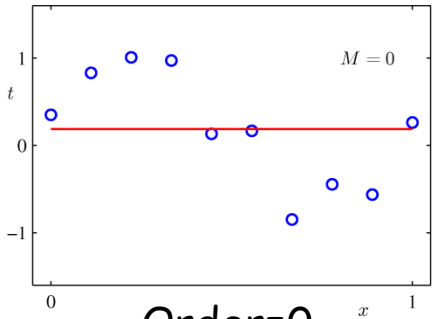
Introduction

- **I want to build a good generalized model**

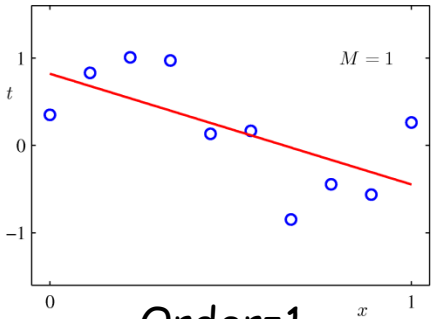


Introduction

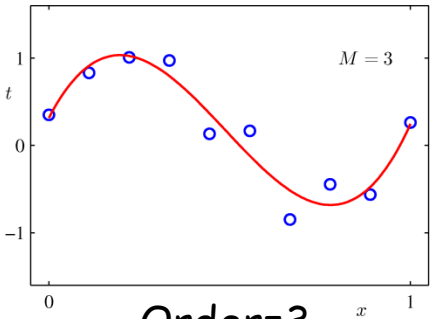
- **Of which model complexity??**
  - As high as possible but not too much to be overfitted
- **How can I choose the best model complexity?**
  - Try one by one and choose the best using Hold-out or CV
  - We usually use this approach



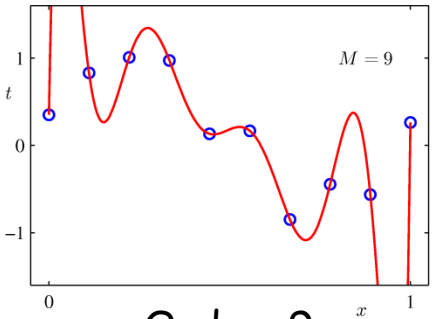
Order=0



Order=1



Order=3



Order=9

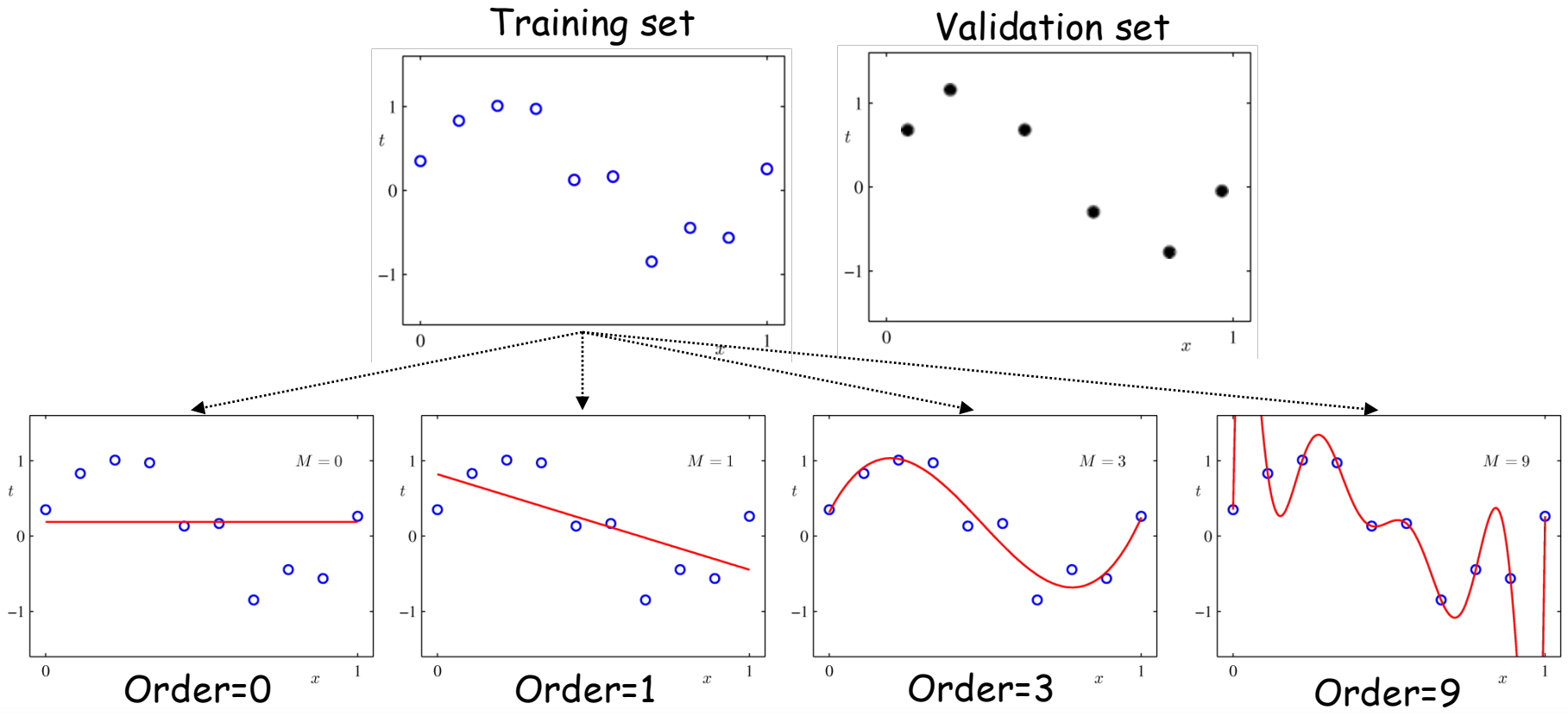
Introduction

- **Choosing the best model**
  - Train many models
  - Evaluate them with Cross-Validation or Hold-out method
  - Choose the best

- **Using Regularization Method**
  - Train one model with a regularization method

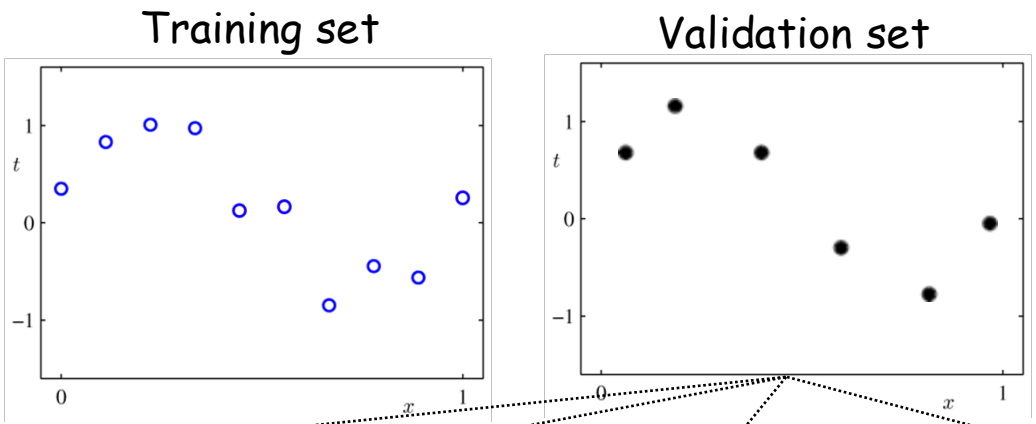
Choosing the Best Model

- **I want to build a good generalized model**
  - Step 1: build several models with training set

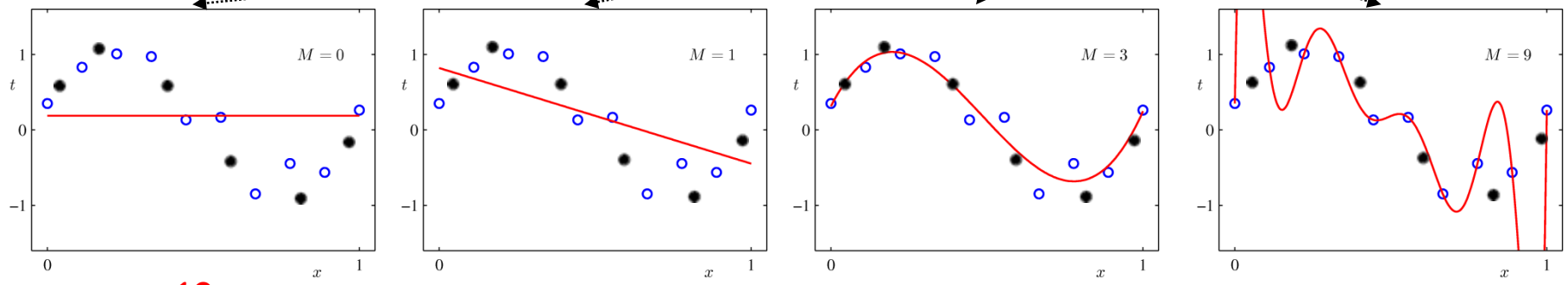


Choosing the Best Model

- **I want to build a good generalized model**
  - Step 2: evaluate the models and choose the best → Time consuming !!!



Validation Error



10 5 1 9

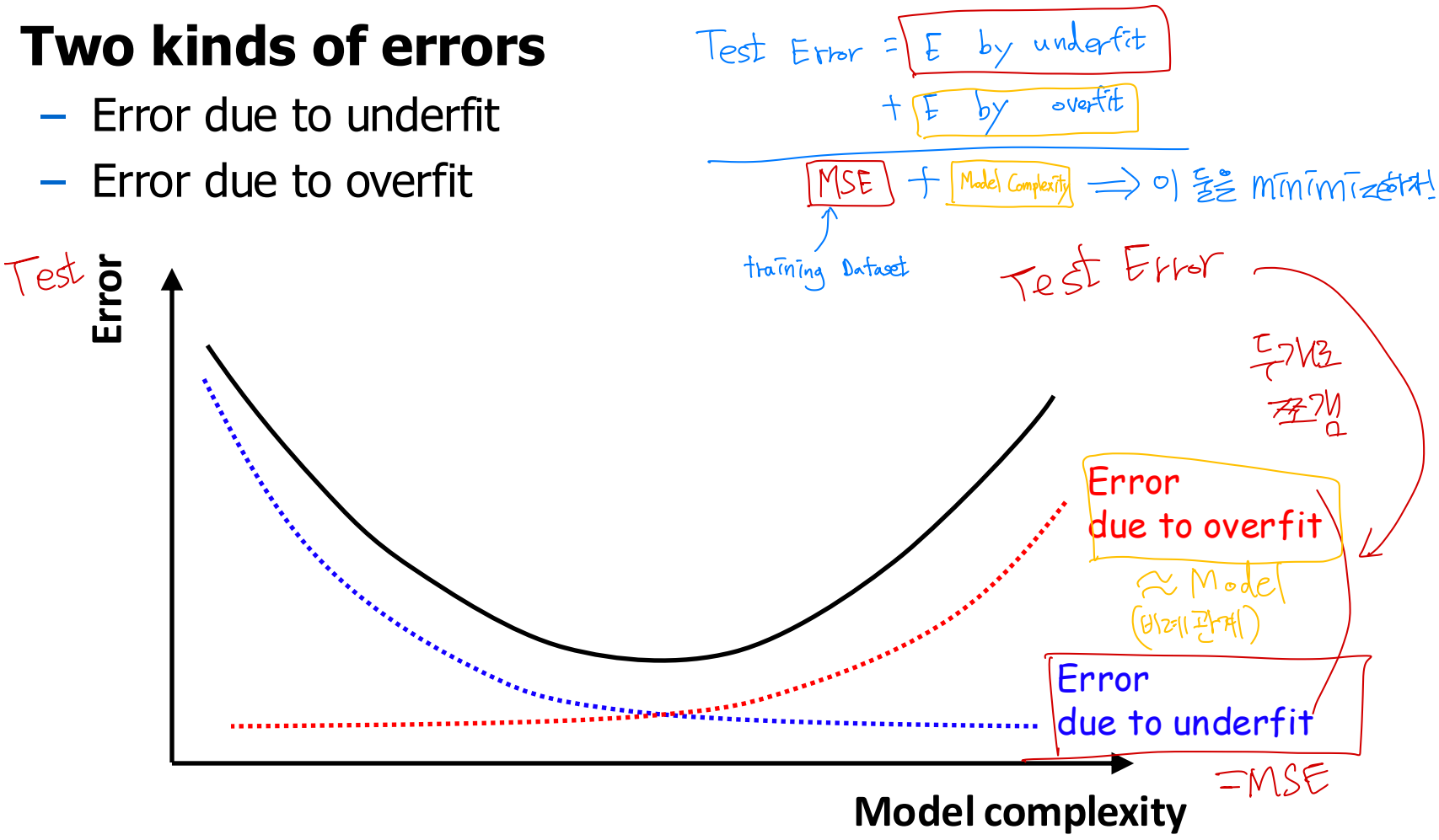
Regularization

- **Regularization**
  - Techniques for automatic adjustment of model complexity
  - Most techniques used while model training.
    Some are applied to the model after training

- **Each ML technique has its own regularization techniques**
  - Additive models: Weight decay
  - DNN: Weight decay, Drop-out …
  - Decision Tree: Pruning, Max-Height Limitation …
  - Naïve Bayes: Smoothing

Weight Decay

- **Two kinds of errors**
  - Error due to underfit
  - Error due to overfit



Weight Decay

- **Two kinds of errors**
  - Error due to underfit ∝ Training Error
    - It may be proportional to the error of training dataset

$$E(\mathbf{w}) = \sum_{(\mathbf{x},y)\in Data} \big(y - f(\mathbf{x})\big)^2$$

  - Error due to overfit ∝ Model complexity
    - It may be proportional to the complexity of models

$$C(\mathbf{w}) = \text{Complexity of the model}$$

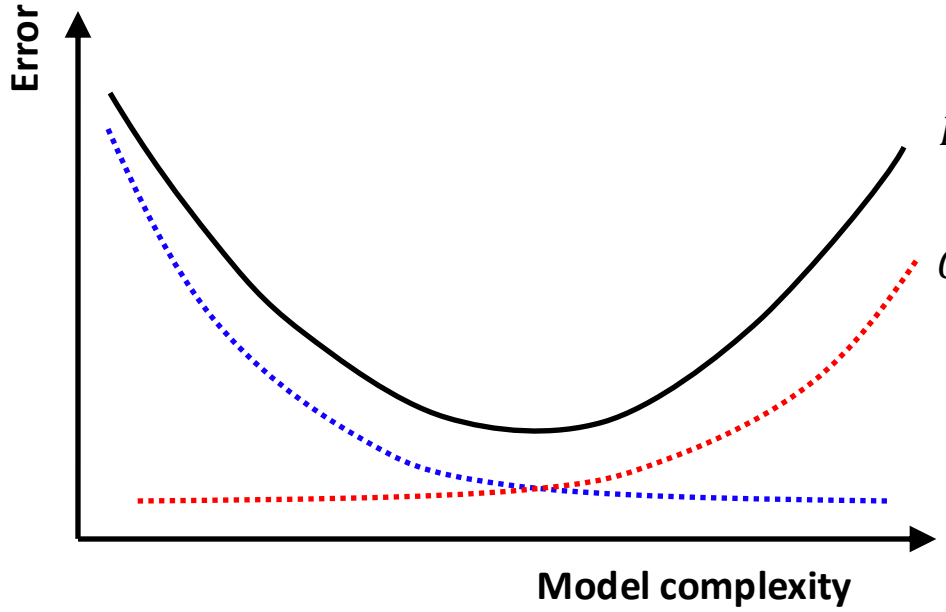
- **If we can minimize the total error, …**

$$\tilde{E}(\mathbf{w}) = E(\mathbf{w}) + C(\mathbf{w})$$

Weight Decay

- **New strategy**
  - We tried to minimize $E(\mathbf{w})$, so we trouble with overfitting
  - We can minimize $\tilde{E}(\mathbf{w})$, we may find the optimal model

But, how can we evaluate the model complexity, $C(\mathbf{w})$?



$$\tilde{E}(\mathbf{w}) = E(\mathbf{w}) + C(\mathbf{w})$$

$$C(\mathbf{w}) = \text{Complexity of the model}$$

$$E(\mathbf{w}) = \sum_{(\mathbf{x},y)\in Data} \big(y - f(\mathbf{x})\big)^2$$

Weight Decay

- **Model Complexity** ⇝ Complexity를 적절하게 설정

Model Complexity increases

$f(x) = w_0$ → 간주 → $f(x) = 0 \cdot x^3 + 0 \cdot x^2 + 0 \cdot x + w_0$

$f(x) = w_1 x + w_0$ → $f(x) = 0 \cdot x^3 + 0 \cdot x^2 + w_1 x + w_0$

$f(x) = w_2 x^2 + w_1 x + w_0$ → $f(x) = 0 \cdot x^3 + w_2 x^2 + w_1 x + w_0$

$f(x) = w_3 x^3 + w_2 x^2 + w_1 x + w_0$ → $f(x) = w_3 x^3 + w_2 x^2 + w_1 x + w_0$

small ← C 함수값 일정하게 증가함 → large

Model Complexity $\propto$ # of Non-zero w's $\propto$ (approximately) $\sum_i |w_i|$ or $\sum_i w_i^2$

- **We may say**

$C(\mathbf{w}) = \sum_i |w_i|$ or $\sum_i w_i^2$

(approximately)

W 성분이 더 많아질수록 C 값이 무조건 커지기 때문

$\sum_i w_i^2$ → 더 많이 사용

$w_2 = 0.001$ 이라면 ≈0이기에 f(x)가 3차지만 0차랑 같아짐 (그래서 approximately) ≠ exactly

Weight Decay

- **Regularization: Weigh Decay**
  - Penalizing the Model Complexity

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} E(\mathbf{w}) + \lambda \cdot C(\mathbf{w})$$

$E(\mathbf{w})$: Training error function, such as MSE

$C(\mathbf{w})$: A function reflecting model complexity

λ: A balancing factor given by experts

  - In the case of Regression
    - Ridge Regression: $C(\mathbf{w}) = \sum_{i=1}^{n} w_i^2$
    - Lasso Regression: $C(\mathbf{w}) = \sum_{i=1}^{n} |w_i|$

Weight Decay

- **Ridge Regression**

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - f(\mathbf{x}_i, \mathbf{w})\right)^2 + \lambda \cdot \sum_{i=1}^{m} w_i^2$$

- **Lasso Regression**

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - f(\mathbf{x}_i, \mathbf{w})\right)^2 + \lambda \cdot \sum_{i=1}^{m} |w_i|$$

How they work ???

Weight Decay

- **How they works?**
  - For example, Ridge Regression

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - f(\mathbf{x}_i, \mathbf{w})\right)^2 + \lambda \cdot \sum_{i=1}^{m} w_i^2$$

$$f(\mathbf{x}, \mathbf{w}) = w_9 x_2^3 + w_8 x_1^3 + w_7 x_2^2 x_1 + w_6 x_2 x_1^2 + w_5 x_2^2 + w_4 x_1^2 + w_3 x_1 x_2 + w_2 x_2 + w_1 x_1 + w_0$$

  - What happens if x_2^3 is less affective to model accuracy?

$$f(\mathbf{x}, \mathbf{w}) = 0 \cdot x_2^3 + 0 \cdot x_1^3 + w_7 x_2^2 x_1 + 0 \cdot x_2 x_1^2 + 0 \cdot x_2^2 + w_4 x_1^2 + 0 \cdot x_1 x_2 + w_2 x_2 + w_1 x_1 + w_0$$

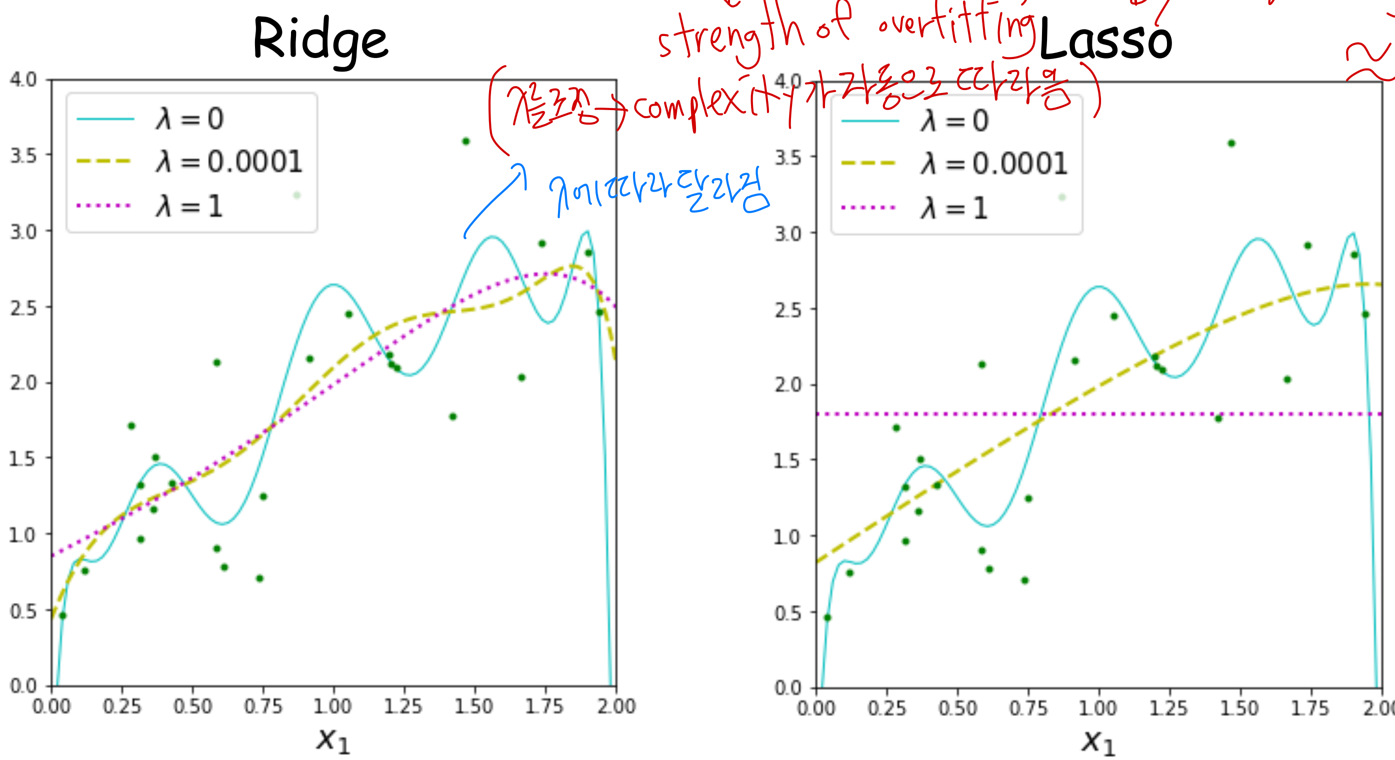
Model is simplified, with less hurt of performance !!

Weight Decay

- **Example**

$$E(\mathbf{w}) + \lambda \cdot C(\mathbf{w})$$

보통은 λmax=1 (데이터셋에 따라 달라짐)
λ=0, MSE만 고려
λ=1, Ridge/Lasso에 따라 다름
strength of overfitting
E by overfitting 측정하기 어려움 ~ model complexity로 측정함
(결조정 → complexity가 자동으로 따라옴)
λ에 따라 달라짐



Weight Decay

$f(x) = w_1 x + w_0$ → $f(x)$가 심플하면 좋은모델 찾기 어렵다

$f(x) = w_3 x^3 + w_2 x^2 + w_1 x + w_0$

↑ 복잡한 경우에 사용하자!

- **L1-Regularization**

$$C(\mathbf{w}) = \sum_{i=1}^{m} |w_i|$$

  - Encourage sparsity by setting weight = 0.
  - Used to select the most informative features.

- **L2-Regularization** — L1보다 더 좋은 방법이다.

$$C(\mathbf{w}) = \sum_{i=1}^{m} w_i^2$$

  - Does not encourage sparsity → small but non-zero weights.
  - Distributes weight across related features (robust).

<robust한 이유>

	$w_3 x^3$	$w_2 x^2$	$w_1 x^1$	w_0	$\sum_{i=1}^{m} \lvert w_i \rvert$ 둘다
	40	0	0	0	40
	10	10	10	10	
L_2	40	0	0	0	$\sum_{i=1}^{m} w_i^2$ < 160 < 400
	10	10	10	10 (stable)	

둘은 구별할 수 있다.

그리고 더 stable한 w를 찾을 수 있다.

안정적인

$MSE + \sum_{i=1}^{m} w_i^2 \Rightarrow L_2$ 정규화