BOAZ

Feature Scaling / Overfitting Underfitting / Regularization

15기 김성용

Index

01 Feature Scaling

Feature Scaling이란?

Feature들의 크기, 범위를 변환시켜주는 방법론

01 Feature Scaling

다음 표와 같은 데이터를 예측에 그대로 사용한다면?

Number of rooms	Years old	Price
3	48	140,200
4	32	300,830
1	73	20,900
6	6	1,140,600

01 Feature Scaling

Feature Scaling 사용 이유

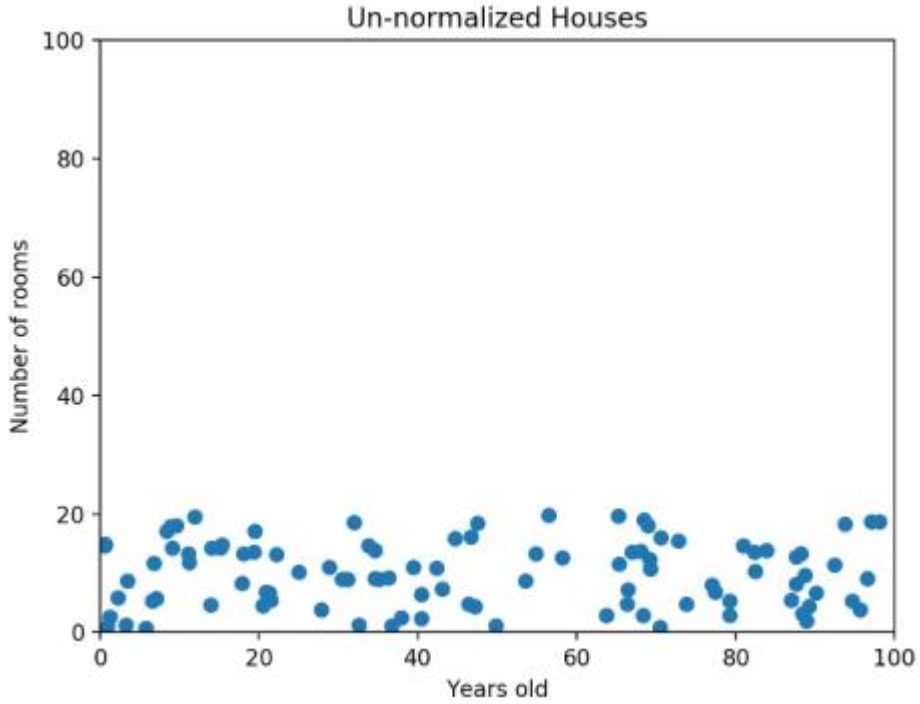
Model output이 큰 features에 의해 좌우되는 것을 방지하기 위해

Feature Scaling
(= Data normalization)

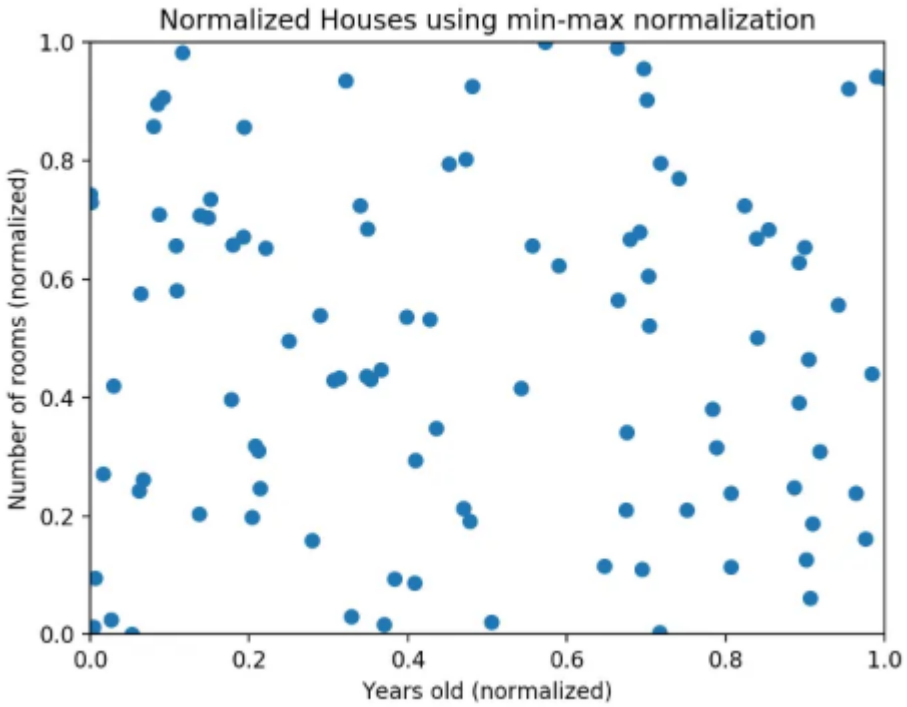
Normalization

Standardization

01 Feature Scaling



Feature Scaling



01 Feature Scaling

Normalization

모든 feature에 대해 각각의 최소값은 0, 최대값은 1로, 그리고 다른 값들은 0과 1 사이의 값으로 변환

$$X = \frac{x - x_{min}}{x_{max} - x_{min}}$$

특징:

- 데이터 군 내에서 특정 데이터가 가지는 위치 파악하기 용이
- 이상치(outlier)를 잘 처리하지 못함

01 Feature Scaling

Standardization

정규분포를 만드는 식과 동일

$$z = \frac{x - \mu}{\sigma}$$

μ = Mean

σ = Standard Deviation

특징:

- 이상치(outlier) 파악에 용이

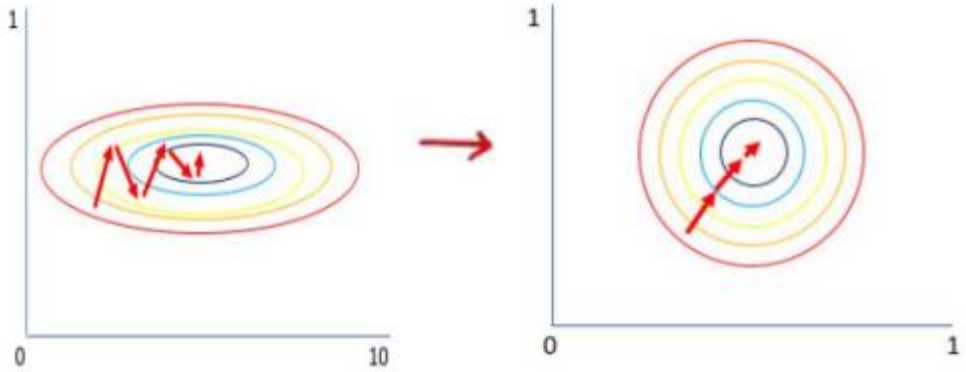
- 2개 이상의 대상이 단위가 다를 때 대상 데이터를 같은 기준으로 볼 수 있음

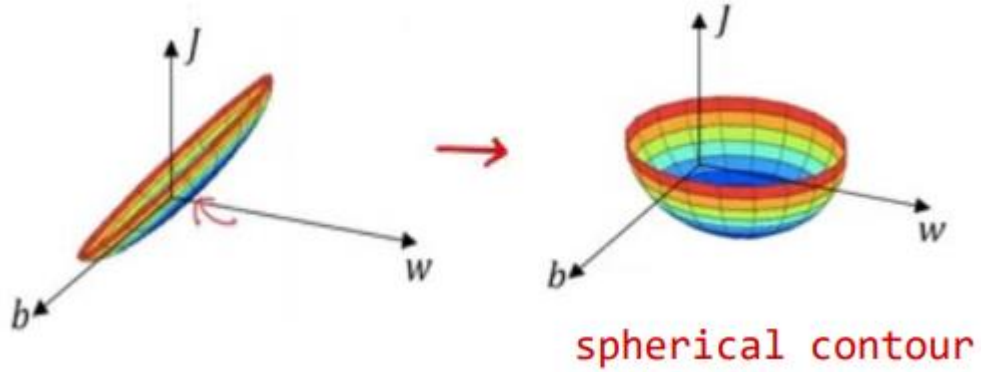
01 Feature Scaling

	표준화(standardization)	정규화(normalization)
공통점	데이터 rescaling	
정의 &목적	데이터가 평균으로부터 얼마나 떨어져있는지 나타내는 값으로, 특정 범위를 벗어난 데이터는 outlier로 간주, 제거	데이터의 상대적 크기에 대한 영향을 줄이기 위해 데이터범위를 0~1로 변환
값의 범위	±1.96(또는 ±2) 데이터만 선택	0~1
공식	$z = \frac{x - \bar{x}}{\sigma}$ (분모가 표준편차)	$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$ (분모가 max값)

01 Feature Scaling

Feature Scaling의 효과





- 모든 Feature의 영향력을 고려할 수 있도록
- (딥러닝에서) 수렴속도(계산비용)의 감소

02 Overfitting vs Underfitting

Underfitting

: 많은 공통특성 중 일부 특성만 반영하여,
too bias하게 train되어 새로운 데이터도 막 예측해버리는 모델

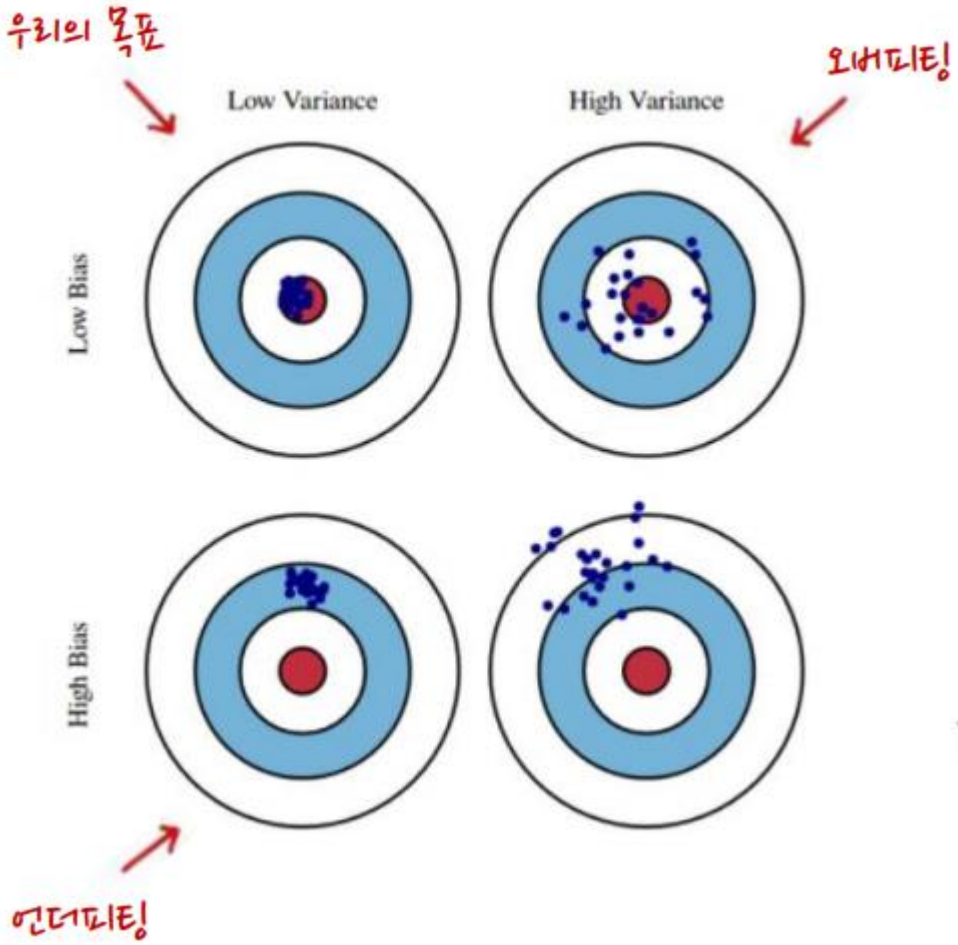
-> 학습이 덜 됨

Overfitting

: 많은 공통특성 외에 지엽적인 특성까지 반영하여,
high variance하게 train되어 새로운 데이터에 대해서는 예측하지 못하는 모델

-> 학습이 너무 과하게 이루어짐

02 Overfitting vs Underfitting



실제값에서 멀어진 척도

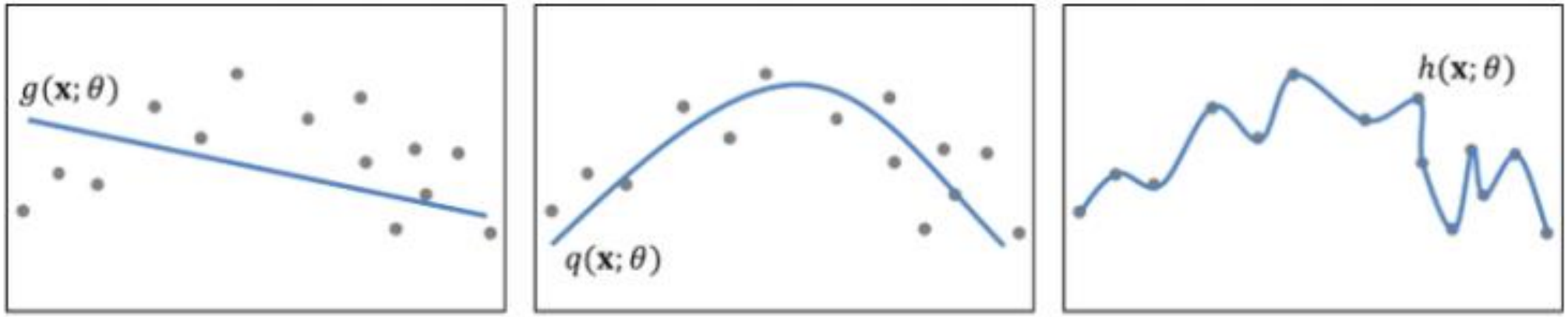
$$Bias = E[f^{pred}(x)] - f(x)$$

Trade off !

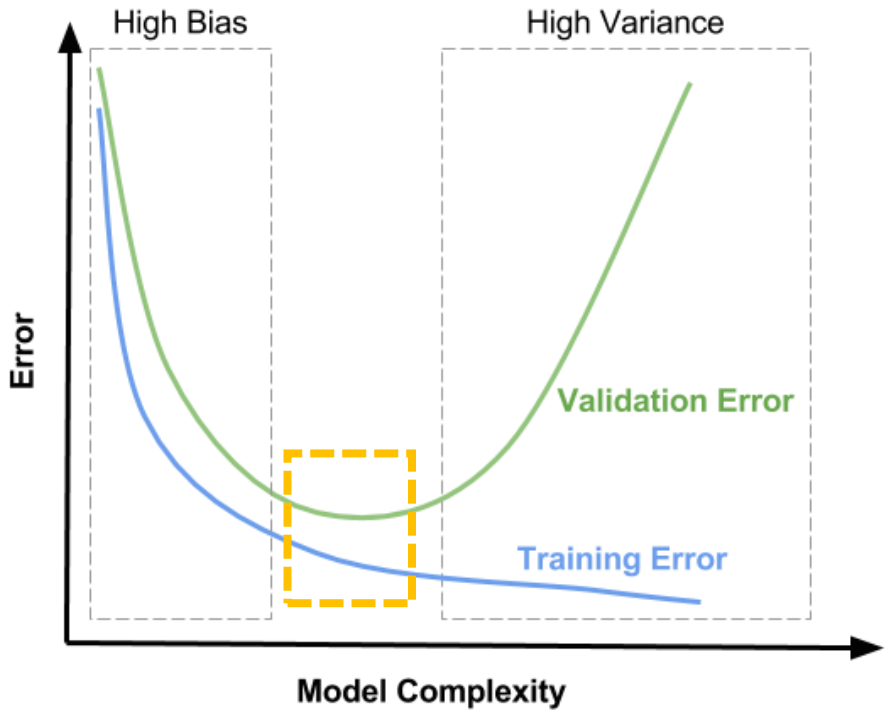
예측값끼리 서로 얼마나 떨어져 있는가

$$Variance = E[f^{pred}(x) - E[f^{pred}(x)]]^2$$

02 Overfitting vs Underfitting



02 Overfitting vs Underfitting



우리의 목표!

02 Overfitting vs Underfitting

Underfitting을 해결하려면

언더피팅은 high bias모델이므로, trade off으로서 bias를 낮추고 variance를 높이자!

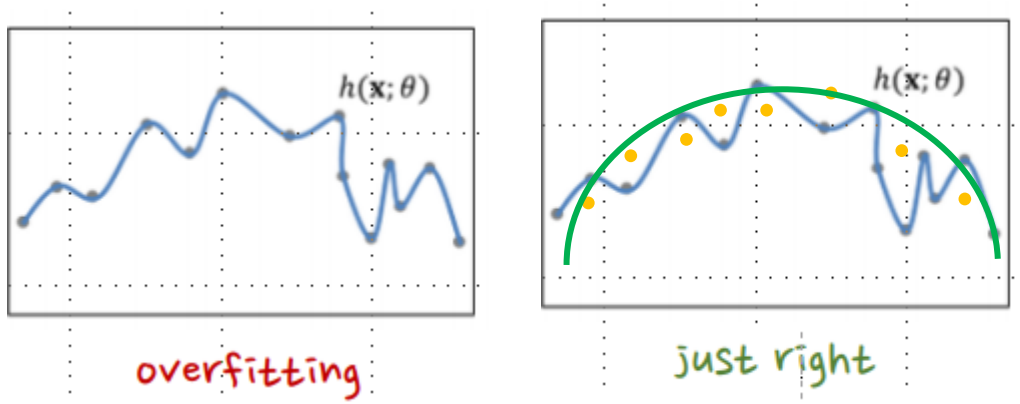
Feature를 더 많이 반영하여, variance 높이기

02 Overfitting vs Underfitting

Overfitting을 해결하려면

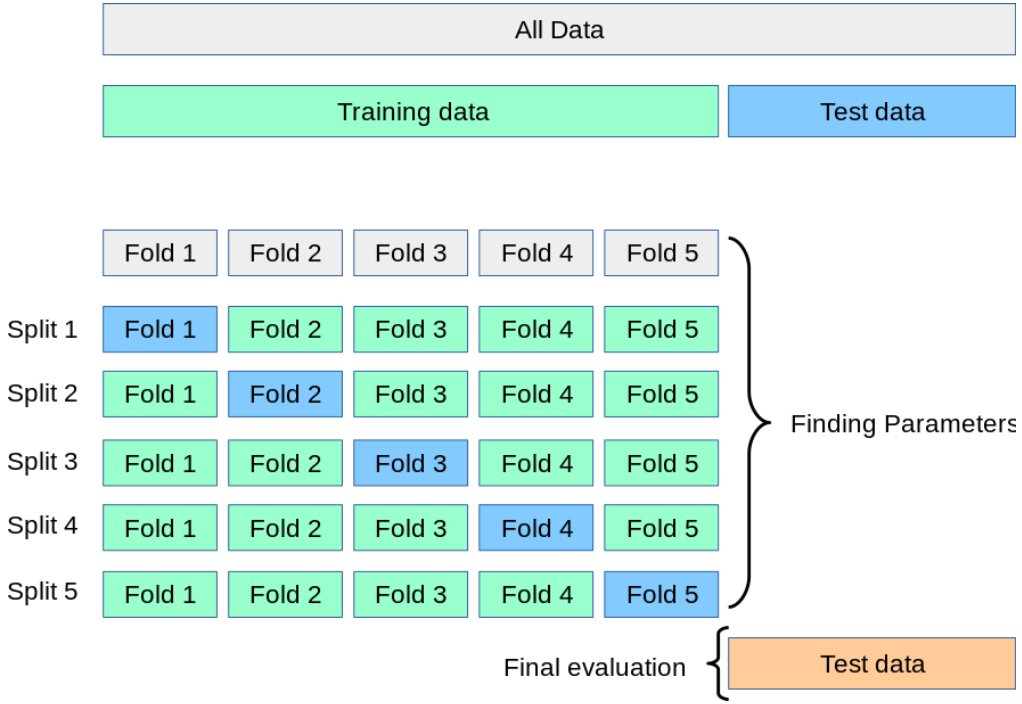
오버피팅은 high variance모델이므로, trade off으로서 bias를 높이고 variance를 낮추자!

1. Feature 수 줄이기
2. 더 많은 데이터 모으기
3. Cross validation 사용
4. Early Stopping, Dropout (딥러닝)
5. Model에 제약걸기(L1 regularization, L2 regularization)



03 Rgularization

Cross validation



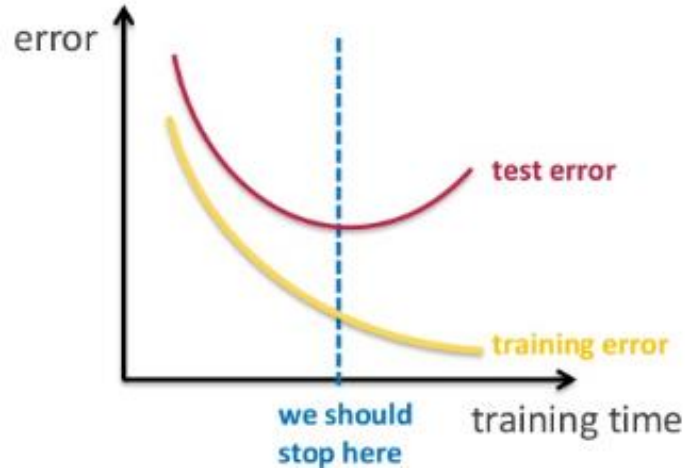
전체 데이터셋을 K개의 subset으로 나누고 K번의 평가를 실행.
(이때 test set의 중복 없이 바꿔가며 평가함.)

장점:
- Train에 들어가는 데이터셋이 계속 바뀌기 때문에
  오버 피팅 방지 가능

단점:
- 모델 훈련/평가 시간이 오래 걸림

03 Rgularization

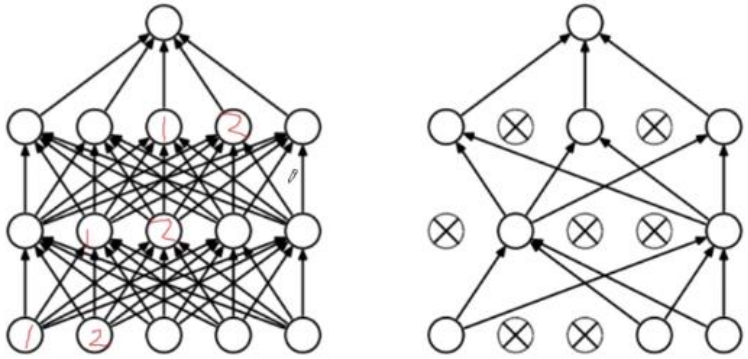
Early stopping, Dropout



Early Stopping

어느 시점에서 train set의 accuracy는 올라가나, validation set의 accuracy는 멈추거나 낮아지는 지점이 옴

validation set의 accuracy가 더 이상 올라가지 않을 때 stop하는 것이 Early Stopping!


(a) Standard Neural Net

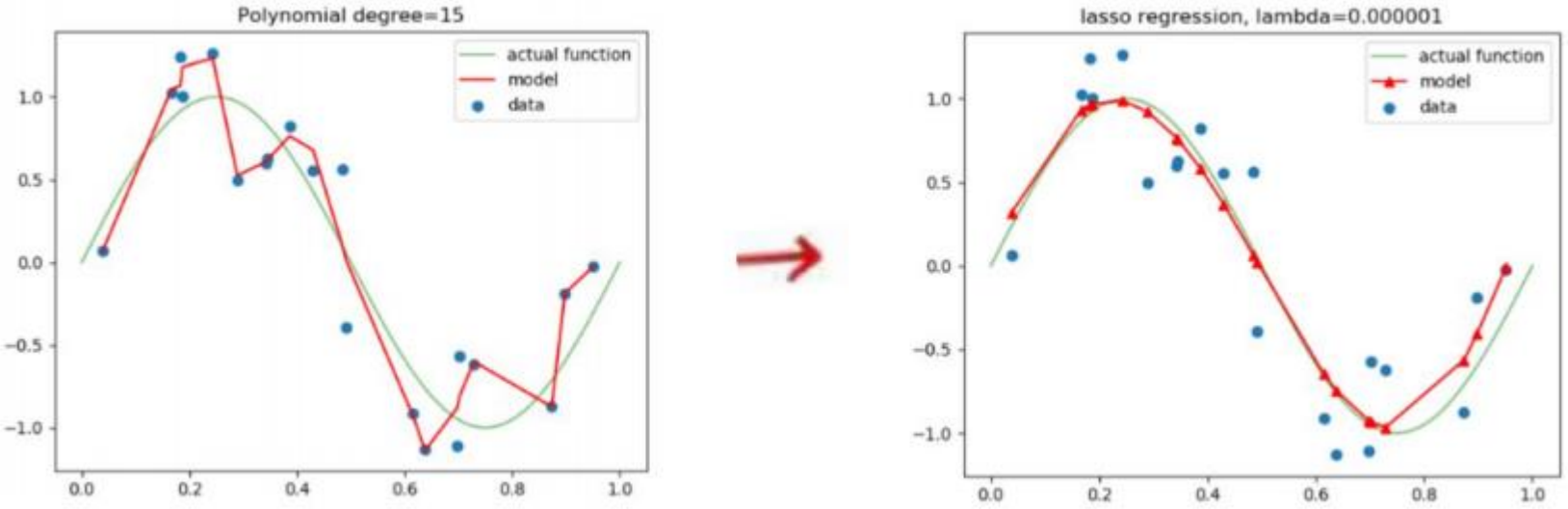
(b) After applying dropout.

Dropout(딥러닝)

Train 과정에서 몇 개의 뉴런을 쉬도록 하여 variance를 낮춤.
몇 번 쉬는 과정에서 overfitting을 막아줌.

03 Rgularization

L1, L2 Regularization



선형회귀 계수(weight)에 대한 제약 조건을 추가함으로써
overfitting을 막는 방법

n개의 parameter가 있을때, 일부 parameter를 작은 값으로 만들어 식을 단순하게!

03 Rgularization

L1 Regularization(Lasso)

$$RSS_{LASSO}(\hat{\beta}) = argmin_{\beta}\left[\frac{1}{2n}\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j|\right]$$

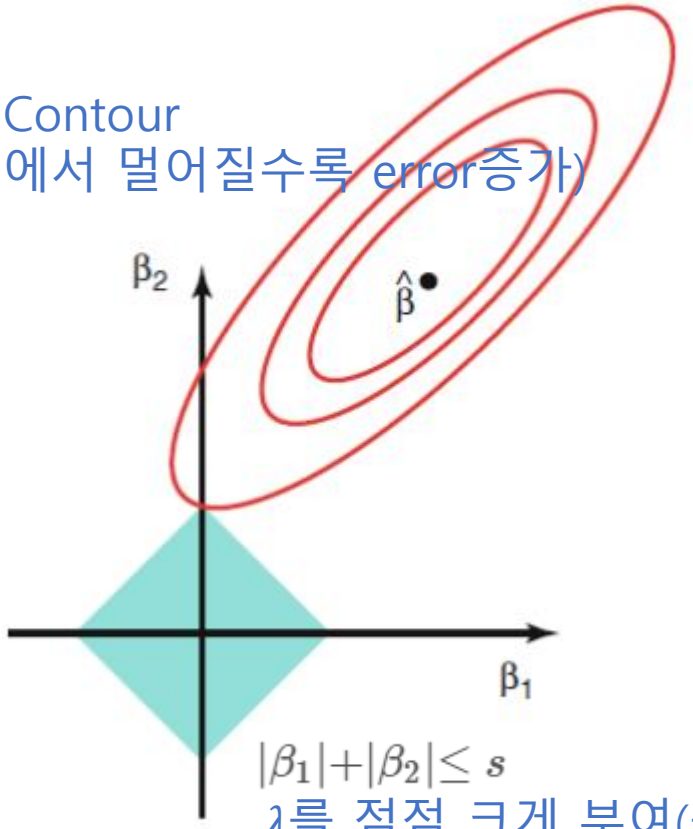
MSE

Penalty

- MSE와 penalty항의 합을 최소로 만드는 값을 찾는 것
- λ는 페널티의 효과 조절 파라미터
- 변수 선택(feature selection) 효과

L1 Regularization(Lasso)

MSE Contour
(중심에서 멀어질수록 error증가)



λ를 점점 크게 부여(즉, penalty를 더 가함)

- Lasso의 제약 범위는 사각형
- 다른 계수가 0인 지점에서 쉽게 교점이 생긴다.
- 3차원(변수가 3개)인 경우 다면체

03 Rgularization

L2 Regularization(Ridge)

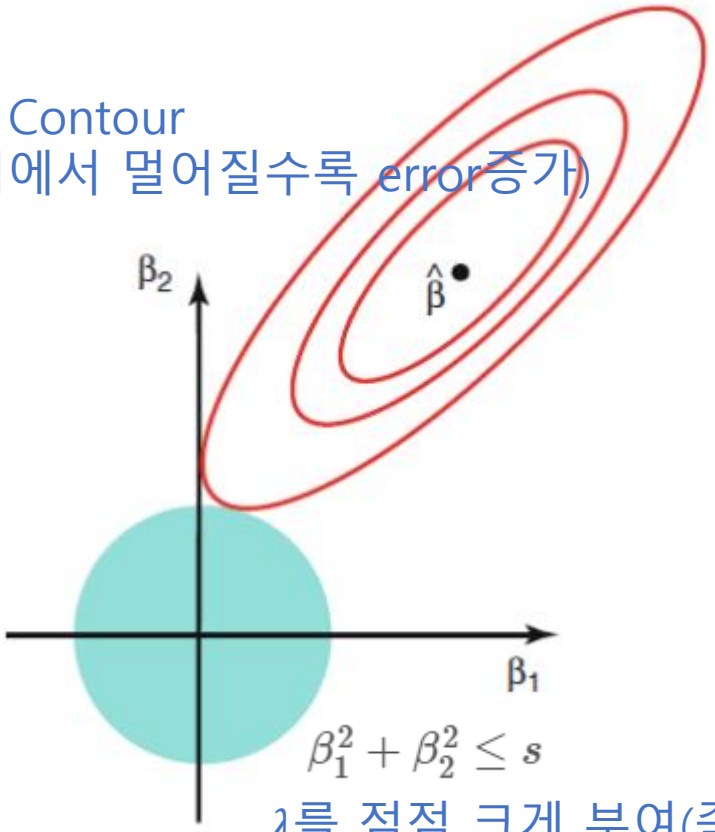
$$RSS_{Ridge}(\hat{\beta}) = argmin_{\beta}\left[\frac{1}{2n}\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{p}\beta_j{}^2\right]$$

MSE penalty

- MSE와 penalty항의 합을 최소로 만드는 값을 찾는 것
- λ는 페널티의 효과 조절 파라미터

L2 Regularization(Ridge)

MSE Contour
(중심에서 멀어질수록 error증가)

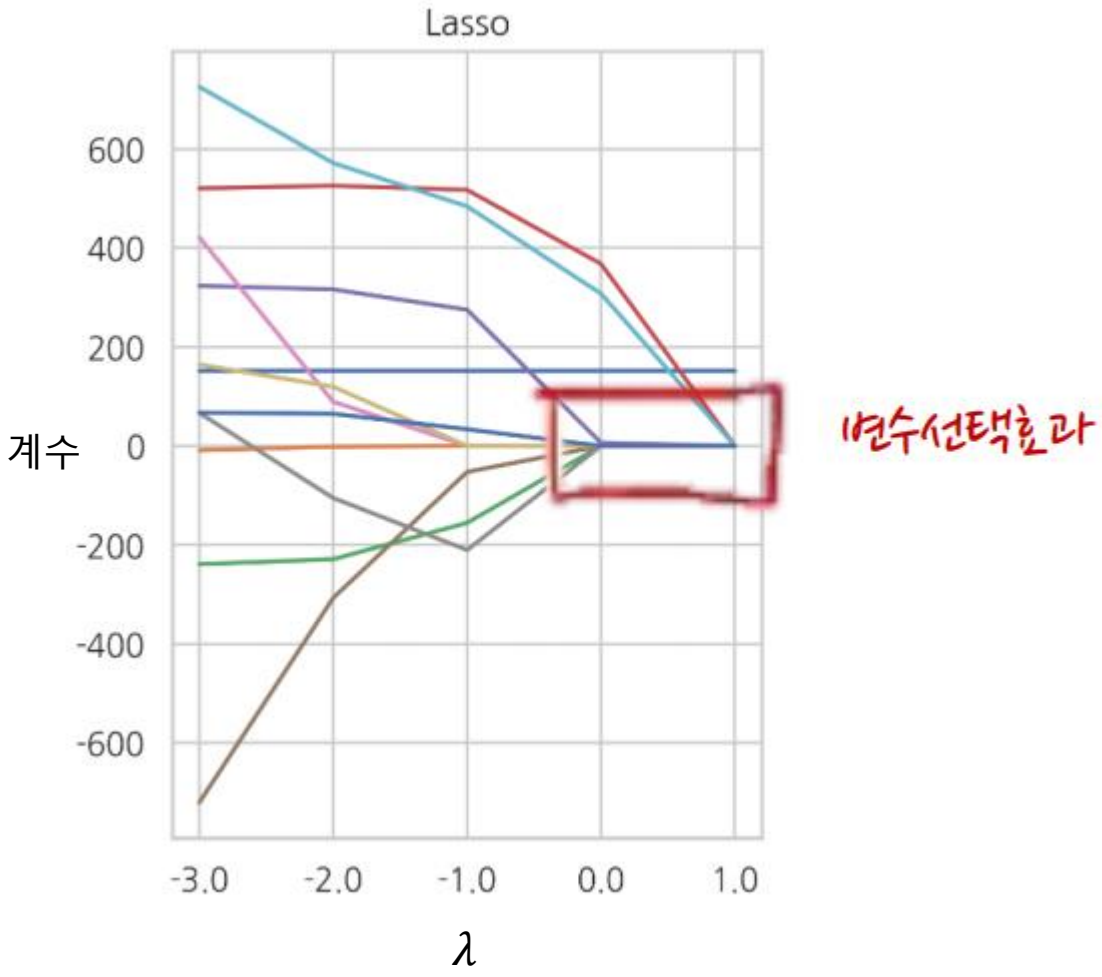


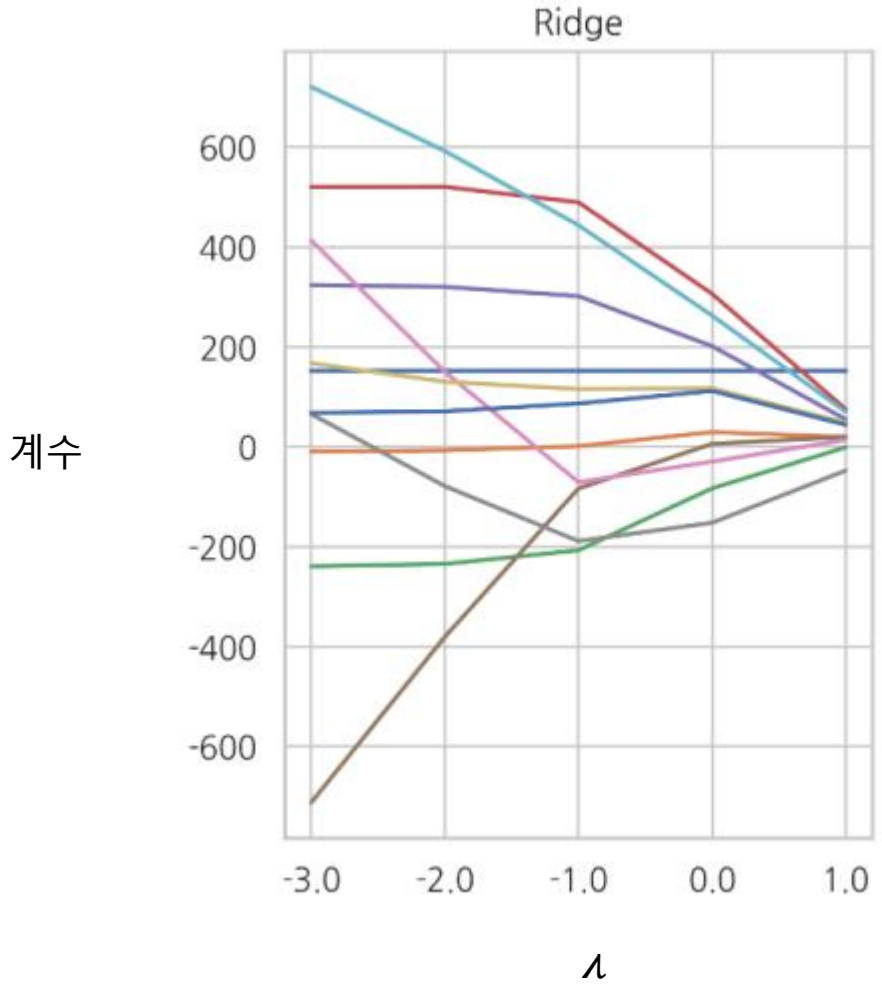
λ를 점점 크게 부여(즉, penalty를 더 가함)

- Ridge의 제약 범위는 원형
- 축에서 교점이 생기기 힘들다.
- 3차원(변수가 3개)인 경우 구

03 Rgularization

L1 Regularization(Lasso) VS L2 Regularization(Ridge)





03 Rgularization

Elastic net(Lasso + Ridge)

$$RSS_{elastic\,net}(\hat{\beta}) = argmin_{\beta}\left[\frac{1}{2n}\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\left[\alpha\sum_{j=1}^{p}|\beta_j| + \frac{(1-\alpha)}{2}\sum_{j=1}^{p}\beta_j{}^2\right]\right]$$

Lasso Ridge

- MSE와 penalty항의 합을 최소로 만드는 값을 찾는 것
- λ는 페널티의 효과 조절 파라미터
- 변수도 줄이고 싶고, 분산도 줄이고 싶은 경우
- (Lasso만 하면 변수가 사라지고 Ridge만 하면 계수는 줄어들지만 변수 선택이 어려움)

03 Rgularization

Ridge	Lasso	Elastic net
변수 간 상관관계 높아도 good 크기 큰 변수 먼저 줄이기	변수 간 상관관계 높으면 성능 bad 중요하지 않은 변수 먼저 없애기	변수 간 상관관계 반영 모두가능

Thank you