Machine Learning

-7주차 실습 과제-

chap.2 모델 검증 및 평가

전자공학과

2022144007

김의진

#과제 1

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)

	Augmentation 하는 이유	
학습시킬 데이터 늘려서 일반화 성능 올리기 위함	Overfitting을 막기 위함	더 이상 데이터 수집 어려울 경우 데이터를 더 얻기 위함

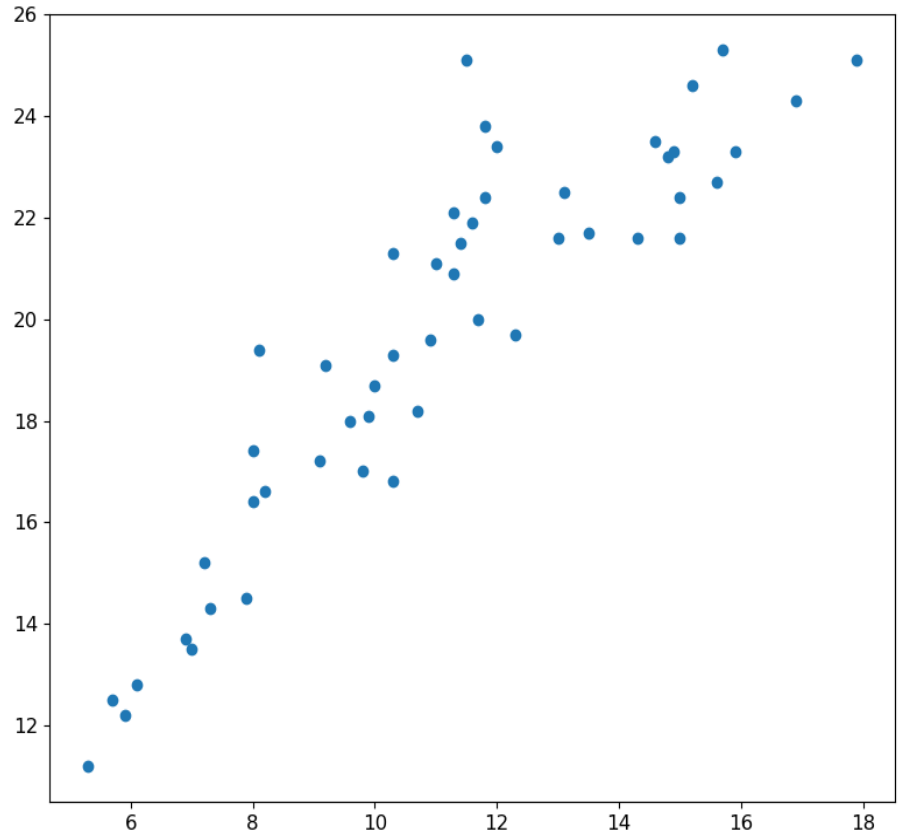
방법
회전
확대/축소
이동
Noise 추가
등등

나는 random noise 발생시키는 것으로 augmentation 할 것임

=> 보다 쉽게 구현할 수 있기 때문

#과제 1

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)



1) 왼쪽 그림과 같은 데이터 있음
2) 데이터마다 그 주위에 일정 노이즈 범위 설정
3) 그 범위안에 random value 20개 만들어줌

-> 데이터 50개 → 데이터1000개로 증강됨

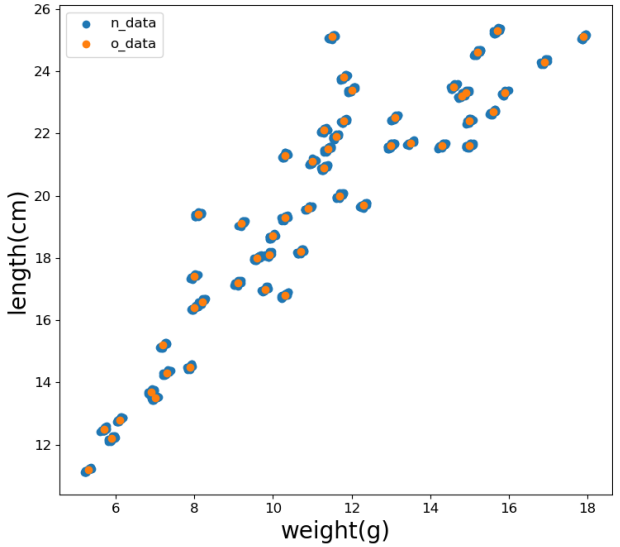
```
Nx = np.random.rand() * (noise) + M[i, 0]
Ny = np.random.rand() * (noise) + M[i, 1]
Nx = np.random.rand() * (-noise) + M[i, 0]
Ny = np.random.rand() * (-noise) + M[i, 1]
```

Noise 사방으로 생기게 범위 설정해줌

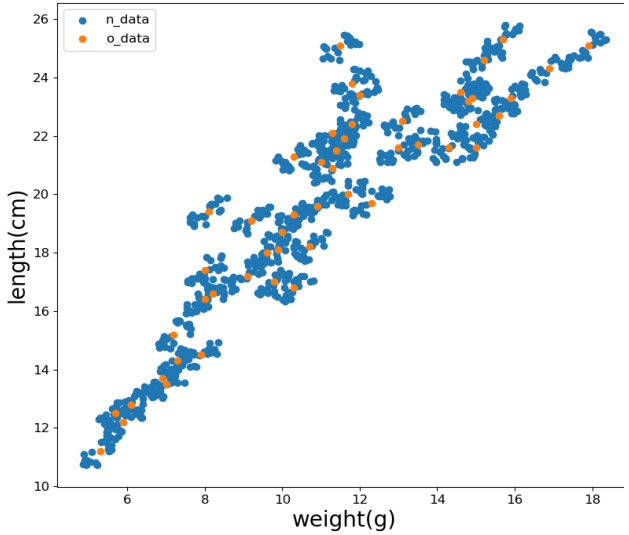
#과제 1

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)

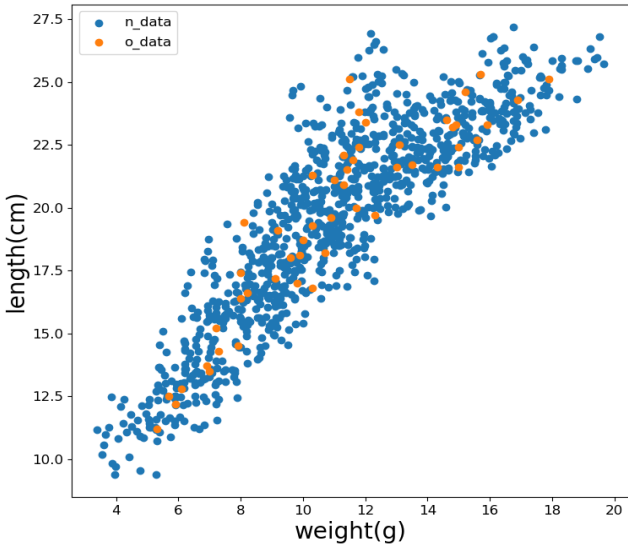
Noise 크기: 0.1



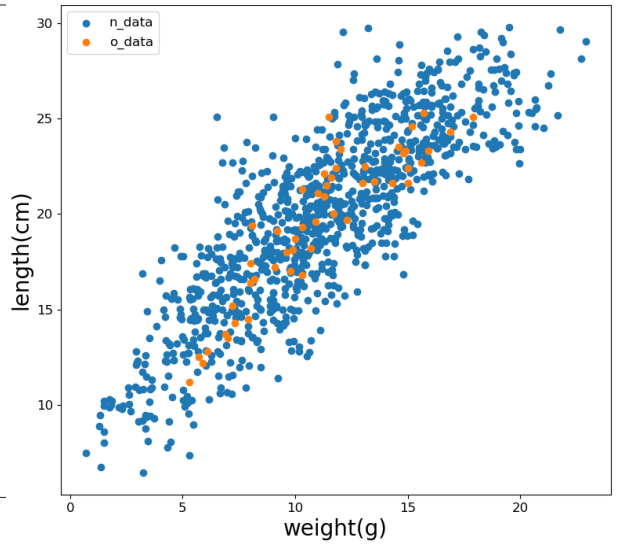
0.5



2



5



Noise의 범위가 커질수록 더 넓게 분포함.

BUT 1) noise범위가 너무 커지면 원래 데이터의 특징과 다른 불필요한 특징을 가지게 돼 잘못된 학습을 하게 됨 → 적절한 noise 범위 설정 필요함

2) noise 범위 너무 작으면 원래 데이터와 차이 없어 다양성이 적으며 쓰레기 데이터를 얻게 됨

#과제 1

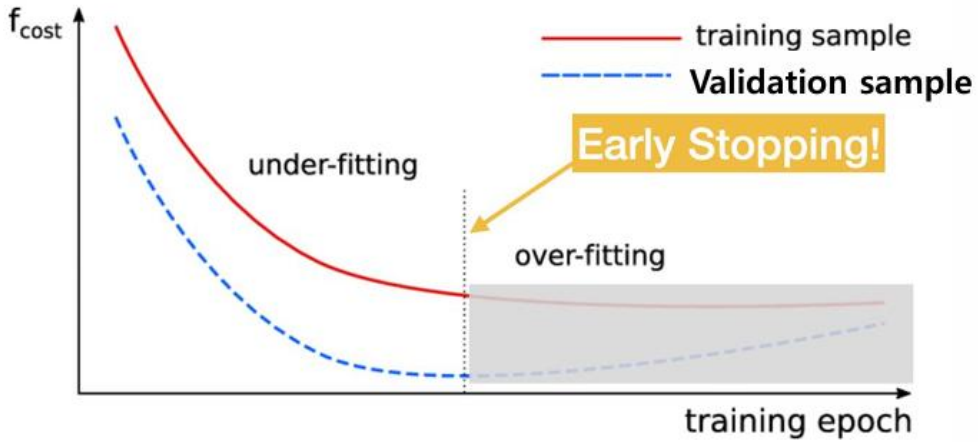
2) 사용자 지정 함수를 활용하여, 보유한Data set을 사용자의 입력비율에 따라 Training set, Validation set, Test set로 분할해주는 함수를 구현하고, 5:3:2로 분할된 데이터를 하나의 그래프에 나타내라

Original set 으로만 학습하면 Overfitting 언제 일어나는지 알 수 없음

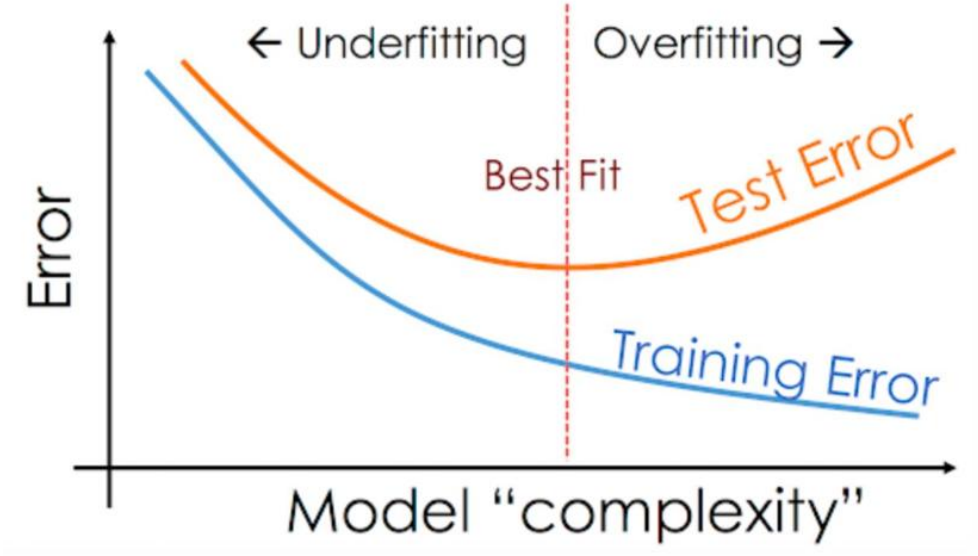
→ Training, validation, testing set으로 나눠서 overfitting 일어나는 때를 봄

#과제 1

2) 사용자 지정 함수를 활용하여, 보유한Data set을 사용자의 입력비율에 따라 Training set, Validation set, Test set로 분할해주는 함수를 구현하고, 5:3:2로 분할된 데이터를 하나의 그래프에 나타내라



경사하강법으로 모델 만들었을 때 최적의 weight 찾기 위해 만듦

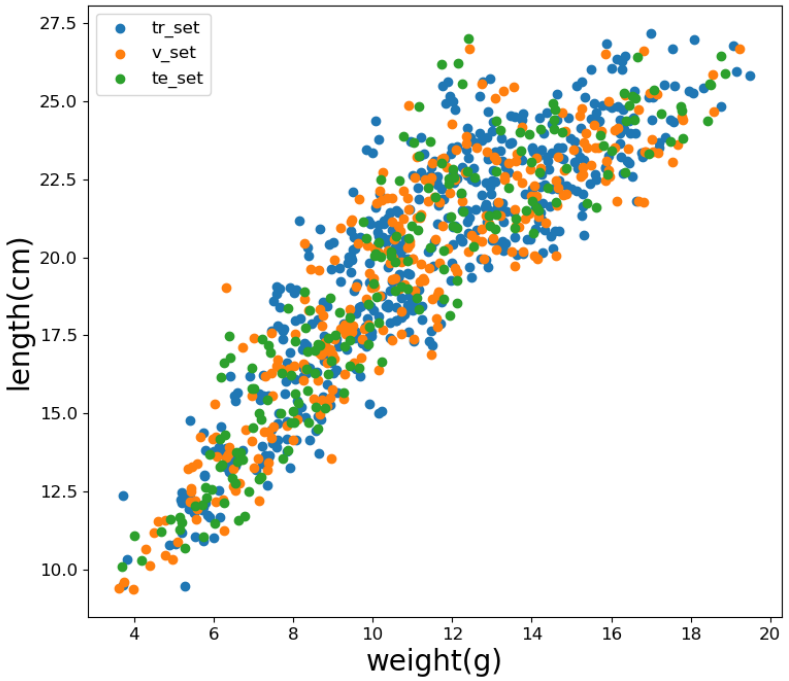


인공신경망에서 weigh수가 늘어날수록, 우리가 하는 실습에서는 basis function이 늘어날수록 complexity 증가
→ 최적의 basis 개수 찾기.

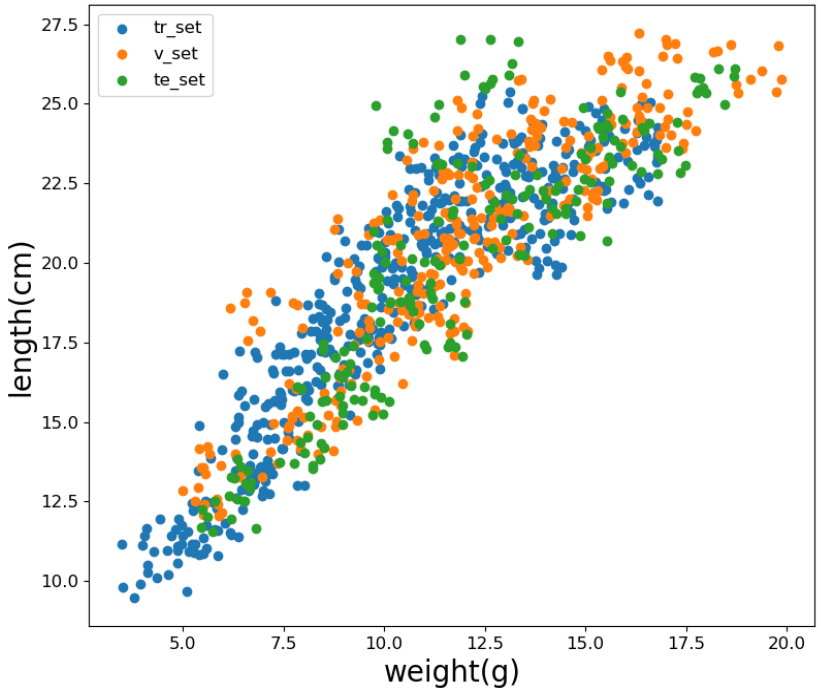
#과제 1

2) 사용자 지정 함수를 활용하여, 보유한Data set을 사용자의 입력비율에 따라 Training set, Validation set, Test set로 분할해주는 함수를 구현하고, 5:3:2로 분할된 데이터를 하나의 그래프에 나타내라

데이터 셔플 후 분할 했을 때



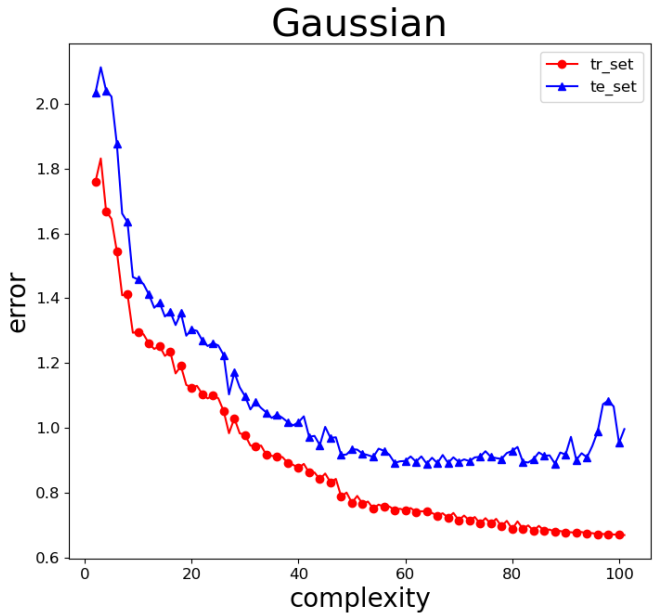
데이터 셔플 안하고 분할 했을 때

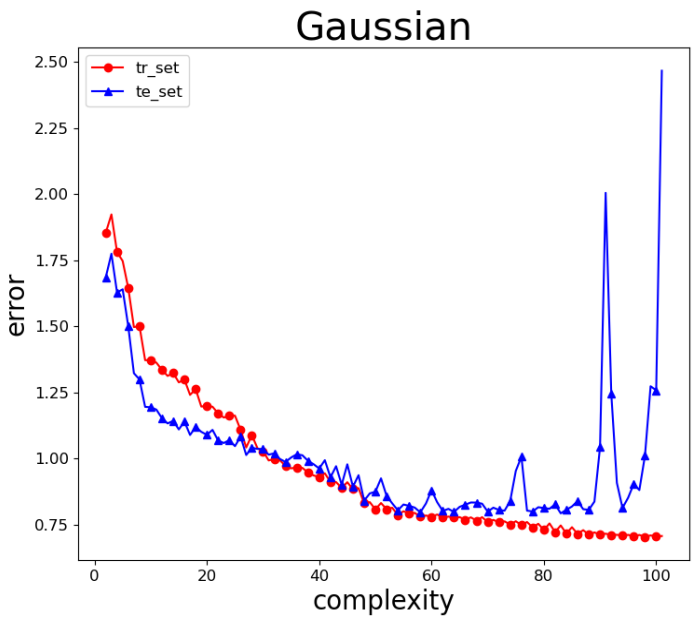


데이터 셔플 안하면 오른쪽 그림과 같이 일정 분포로 training, validation, testing set이 안 나누어질 수 있음 → 데이터 셔플의 필요성

#과제 1

3) lin_regression_data_01.csv 데이터에 대해서 8:0:2로 set을 나누고, Chap.1에서 구현한 가우시안 기저 함수 모델 코드를 응용하여 그래프를 그리고, 최적의 K(가우시안 기저 함수 개수) 를 도출하라.





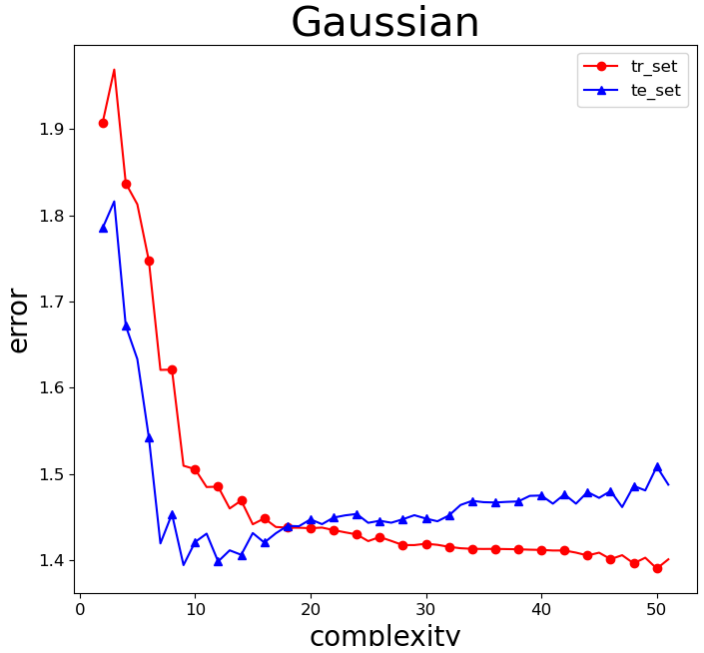
Noise의 범위가 0.1일 때 그래프 2개

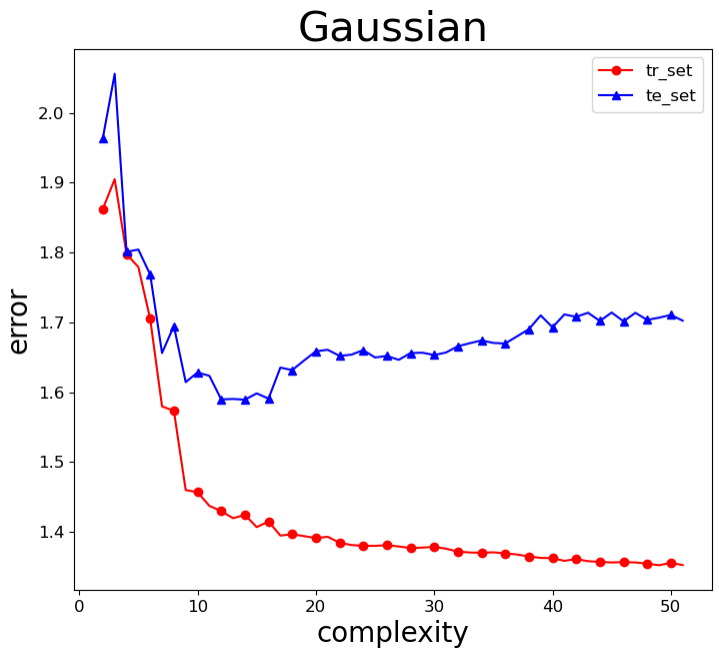
Basis function의 개수가 88개 일때 test set의 그래프가 상승하기 시작

Noise가 0.1일 때 k의 최적 값은 88

#과제 1

3) lin_regression_data_01.csv 데이터에 대해서 8:0:2로 set을 나누고, Chap.1에서 구현한 가우시안 기저 함수 모델 코드를 응용하여 그래프를 그리고, 최적의 K(가우시안 기저 함수 개수) 를 도출하라.





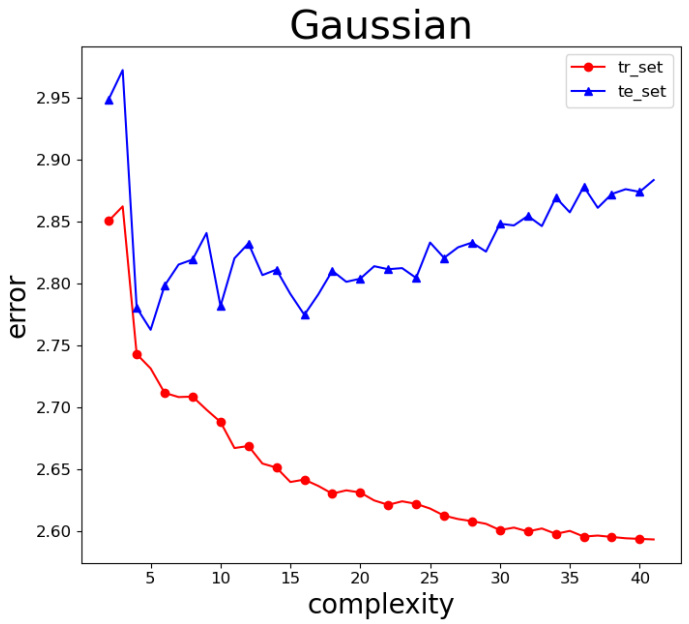
Noise의 범위가 0.5일 때 그래프 2개

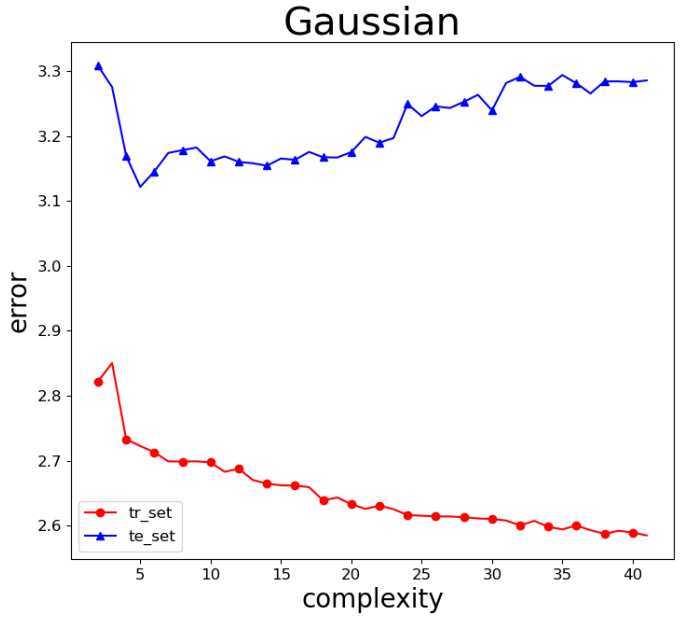
Basis function의 개수가 12개 일 때 test set의 그래프가 상승하기 시작

Noise가 0.5일 때 k의 최적 값은 12

#과제 1

3) lin_regression_data_01.csv 데이터에 대해서 8:0:2로 set을 나누고, Chap.1에서 구현한 가우시안 기저 함수 모델 코드를 응용하여 그래프를 그리고, 최적의 K(가우시안 기저 함수 개수) 를 도출하라.





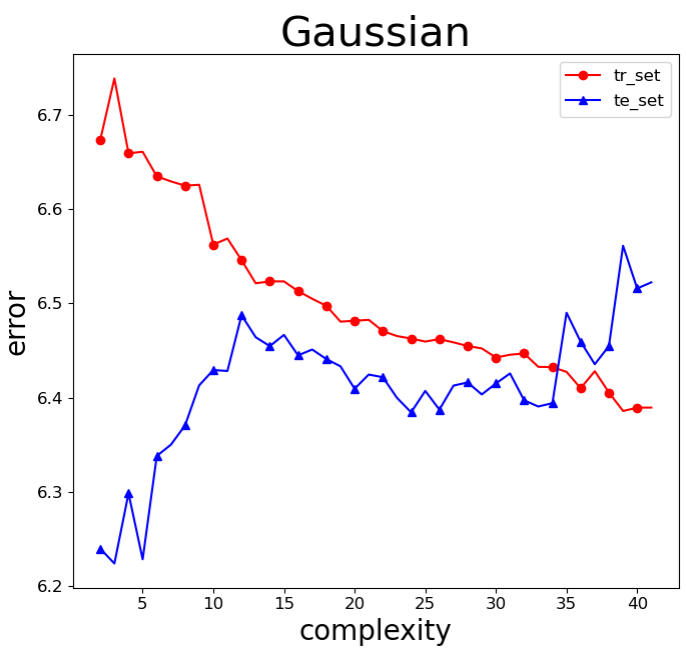
Noise의 범위가 2일 때 그래프 2개

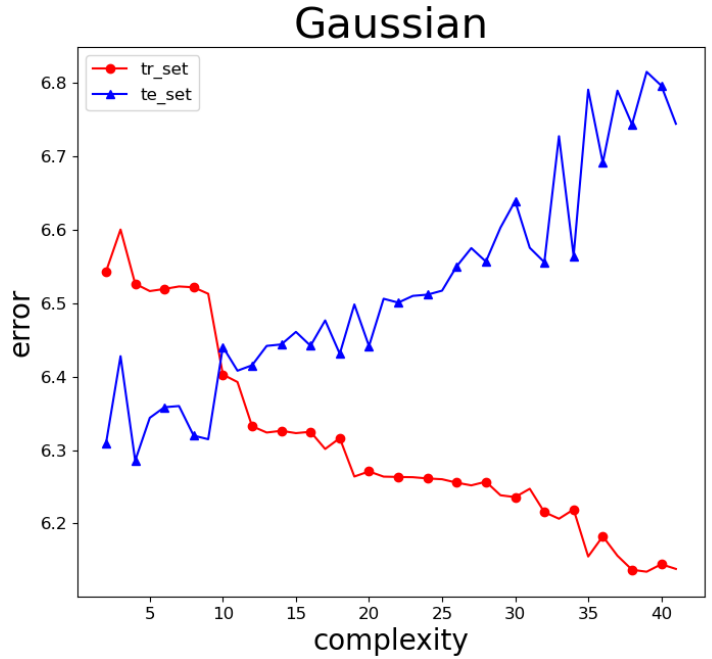
Basis function의 개수가 5개 일 때 test set의 그래프가 상승하기 시작

Noise가 2일 때 k의 최적 값은 5

#과제 1

3) lin_regression_data_01.csv 데이터에 대해서 8:0:2로 set을 나누고, Chap.1에서 구현한 가우시안 기저 함수 모델 코드를 응용하여 그래프를 그리고, 최적의 K(가우시안 기저 함수 개수) 를 도출하라.





Noise의 범위가 5일 때 그래프 2개

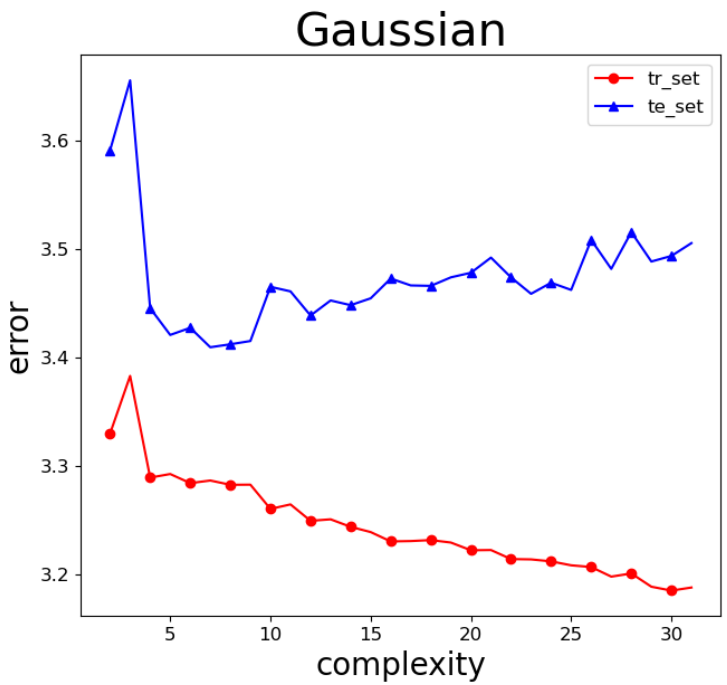
모델이 처음부터 계속 우상향 하는 모습이 보임

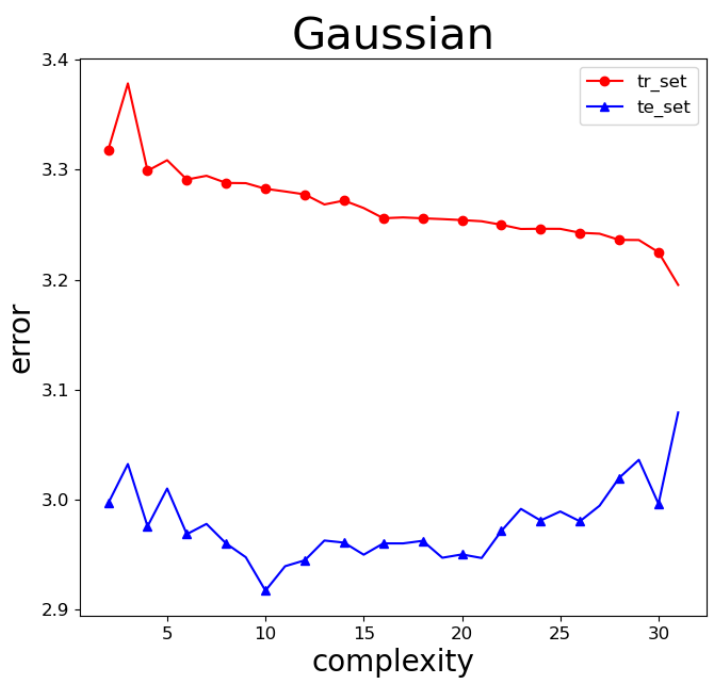
→ Noise 범위가 너무 커져서 원래 특성과는 다른 데이터에 대해 학습하게 돼 test set의 그래프가 우상향함

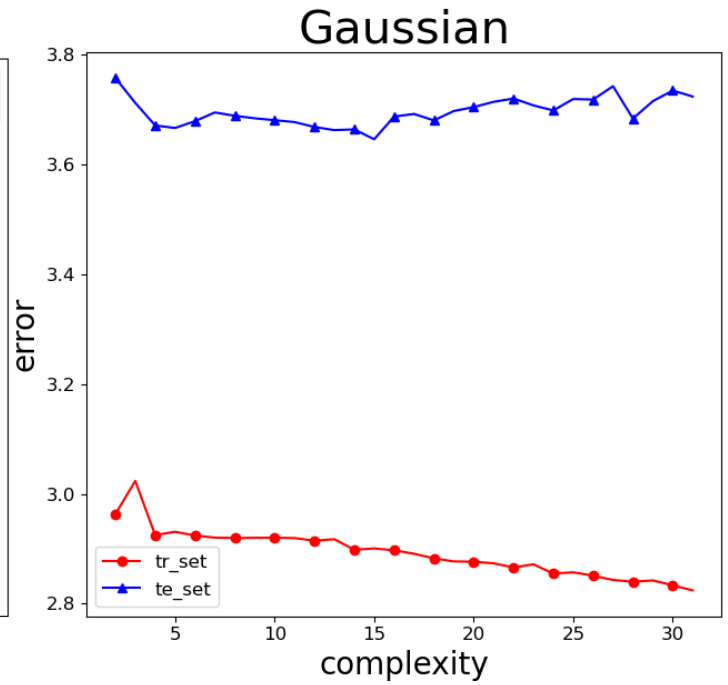
#과제 1

추가 실습 noise를 조절하며 안정적으로 underfitting, overfitting을 확실히 판단할 수 있을 범위를 구하기

Noise가 2일 때 이러한 경향을 띄는 그래프가 많이 나옴 → 안정적으로 학습이 잘 되었다고 볼수 없음

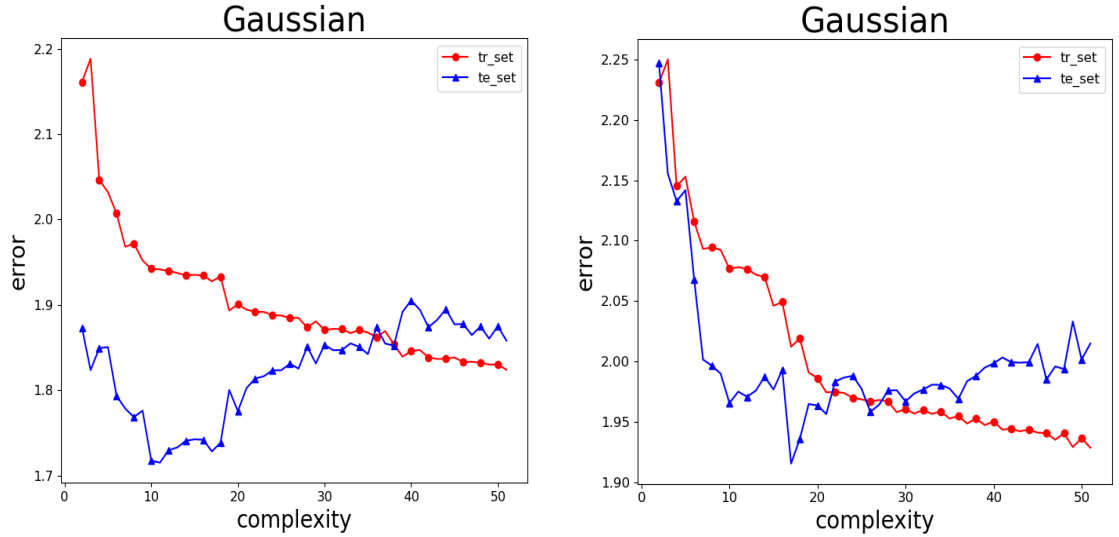


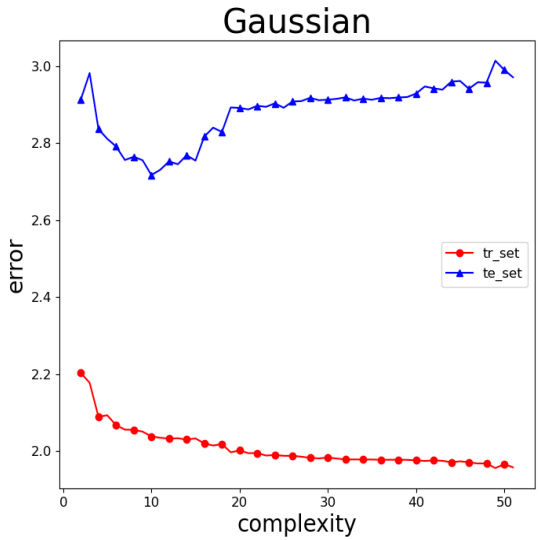


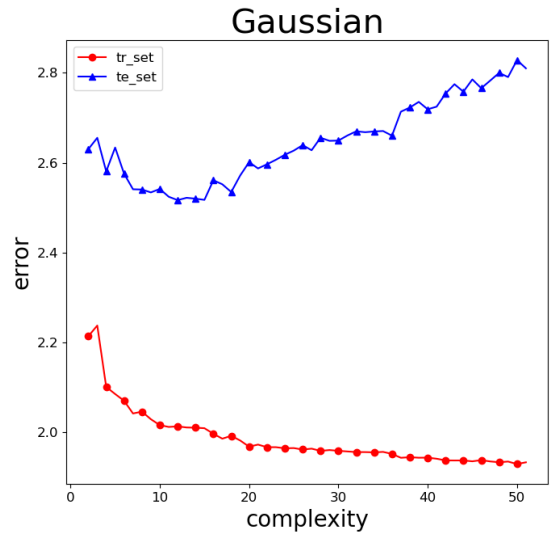


#과제 1

추가 실습 noise를 조절하며 안정적으로 underfitting, overfitting을 확실히 판단할 수 있을 범위를 구하기





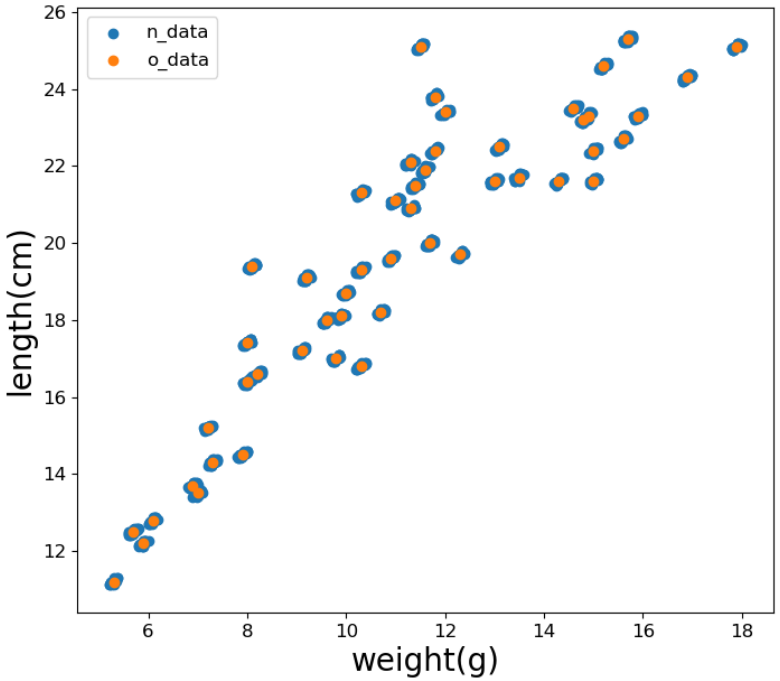


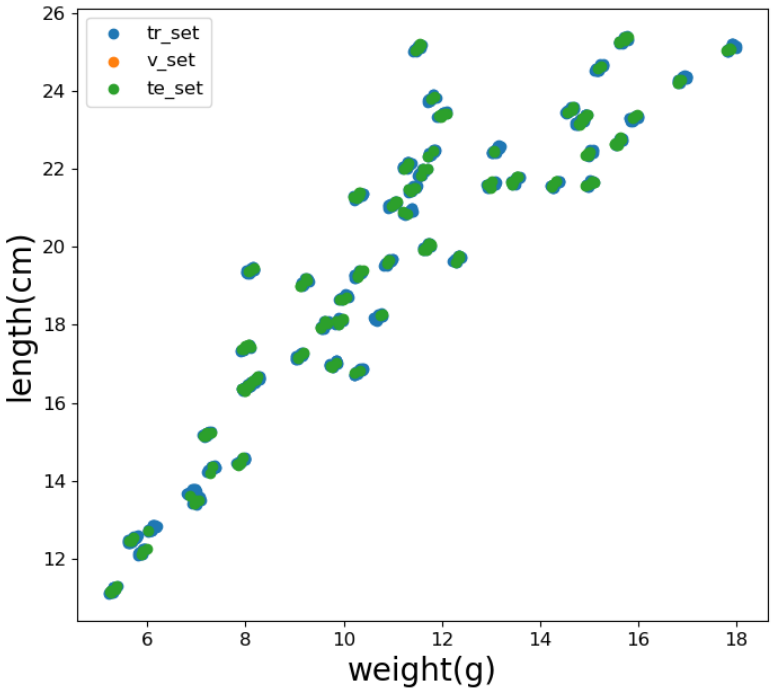
Noise가 1.2가 되어서야 안정적으로 error가 내려가다가 다시 상승하는 곡선이 그려짐

그럼 noise가 작아지면 작아질 수록 학습이 잘 되는 것일까?

#과제 1

추가 실습 noise를 조절하며 안정적으로 underfitting, overfitting을 확실히 판단할 수 있을 범위를 구하기





NO

1) 극단적으로 noise 0.1 설정

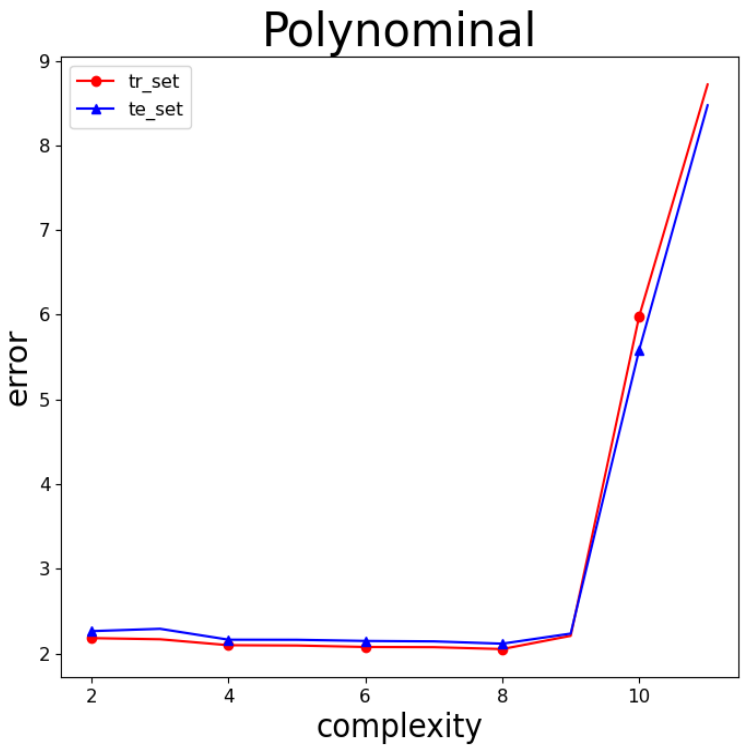
2) 옆의 그림을 보면 원래 데이터와 다른점이 거의 없음

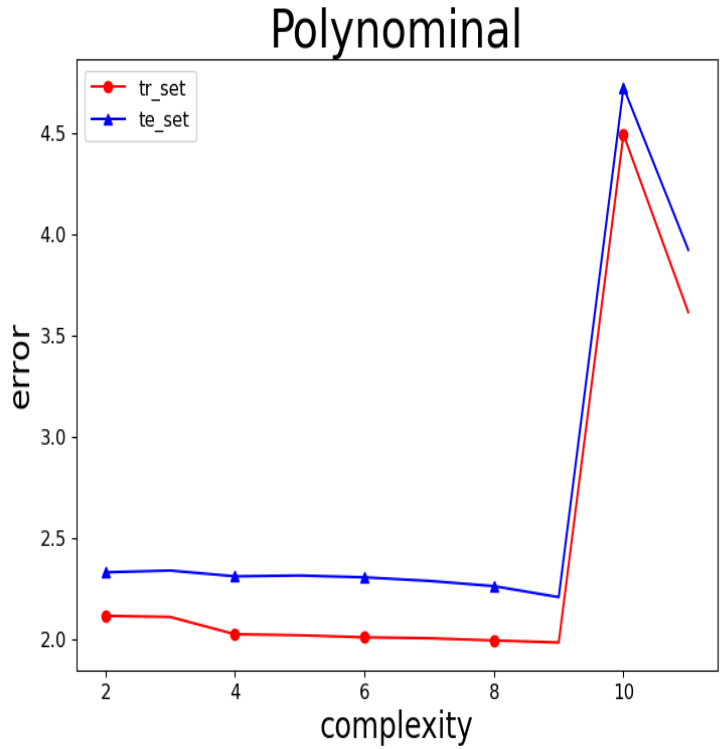
3) 오른쪽 그림을 봤을 때 tr_set과 te_set의 차이가 없음

→다양성 부족, 양질의 데이터가 아님.제대로 된 학습이 될 수 없음

#과제1

추가 실습 polynominal basis function으로 해보기





옆의 그림과 같이 tr_set, te_set 값이 모두 폭발한 후 상승함

1) 미묘하게 하락하는 모습
→ 기저 개수 적을 때는 학습이 되는 모습

2) tr_set, te_set 값 같이 폭발하는 모습
→ 함수의 꼴이 $y = x^k$이기 때문에 k 값이 커지면 쉽게 불안정해짐