

학습곡선

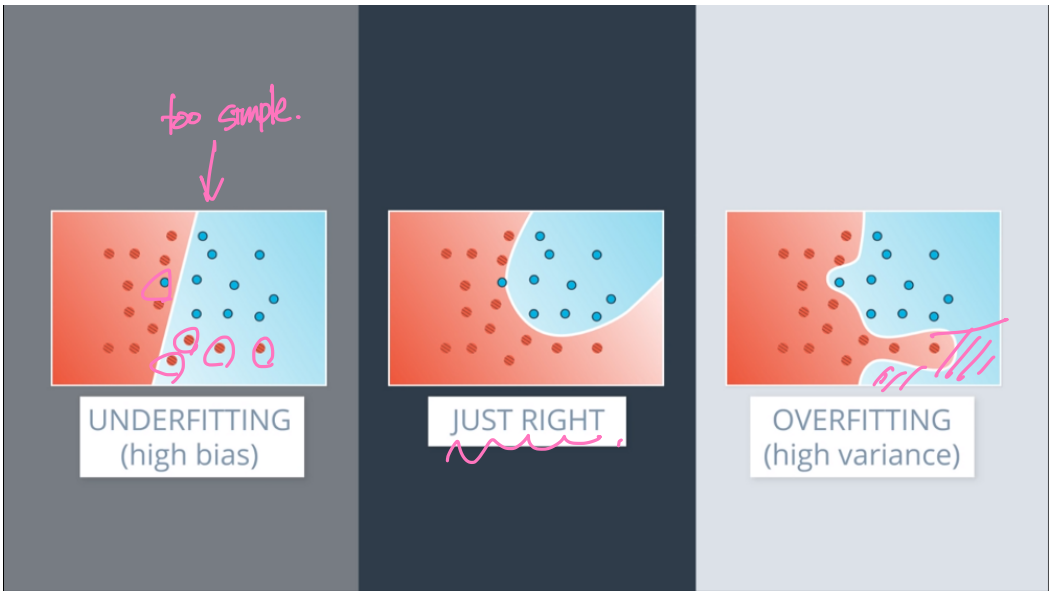
Data가 적다. ~~복잡한~~ model.
Data ↓, model 단순.
~~~~~ 복잡 차와높이 (바→낮아.)  
\* Feature 외갓수 줄임 ✓

## Overfitting/Underfitting

- 과대 적합이란 모델이 훈련 세트에서는 좋은 성능을 내지만 검증 세트에서는 낮은 성능을 내는 경우를 말한다.
- 훈련세트의 정확도가 99% 이고, 검증세트의 정확도가 80% 수준이라면 과대 적합을 의심할 수 있다.  $r^2$ : train : 0.8 , test : 0.6.
- 과소 적합은 훈련세트와 검증 세트의 성능에는 차이가 크지 않지만 모두 낮은 성능을 내는 경우를 의미한다.  $r^2$ : train 0.4 , test 0.3.

## 언제 나타날까?

- 매개 변수가 많고 표현력이 높은 모델  $\Rightarrow$  Overfitting
- 훈련 데이터가 너무 적은 모델
- 



## 분석 방법

- 학습곡선을 통해 과대 적합과 과소 적합을 분석할수 있다.

```
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
history = model.fit(train_X, train_Y, epochs=5, batch_size=16384, validation_split=0.2)

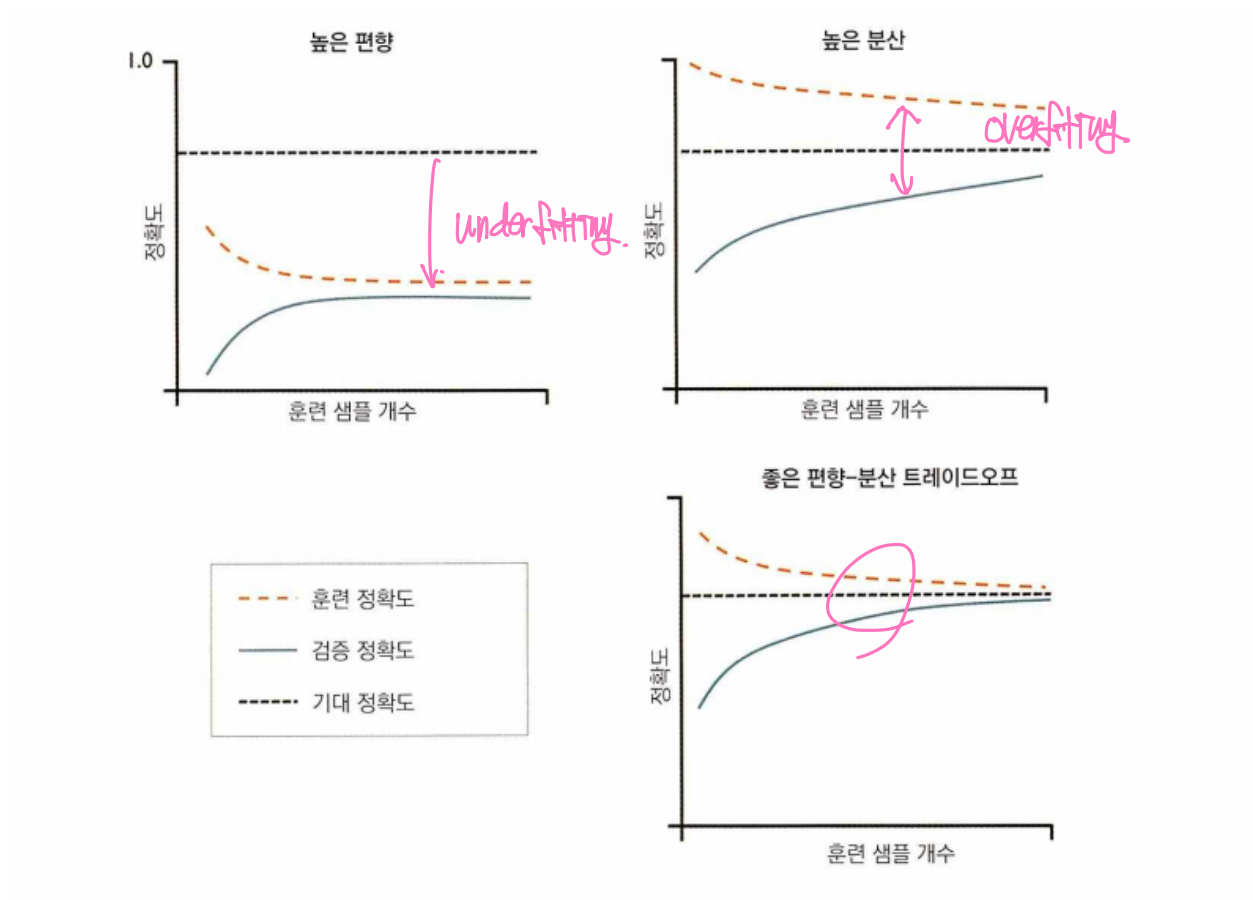
loss = history.history['loss']
val_loss = history.history['val_loss']
plt.figure(figsize=(8, 8))
plt.subplot(2, 1, 1)
plt.plot(acc, label='Training Accuracy')
plt.plot(val_acc, label='Validation Accuracy')
plt.legend(loc='lower right')
plt.title('Training and Validation Accuracy')
plt.xlabel('epoch')
```

```
plt.ylabel('Accuracy')
# plt.ylim([min(plt.ylim()),1])

plt.subplot(2, 1, 2)
plt.plot(loss, label='Training Loss')
plt.plot(val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.title('Training and Validation Loss')
plt.xlabel('epoch')
plt.ylabel('Cross Entropy')
# plt.ylim([0,1.0])

plt.show()
```

## 훈련세트의 크기와 과대적합/과소적합 분석



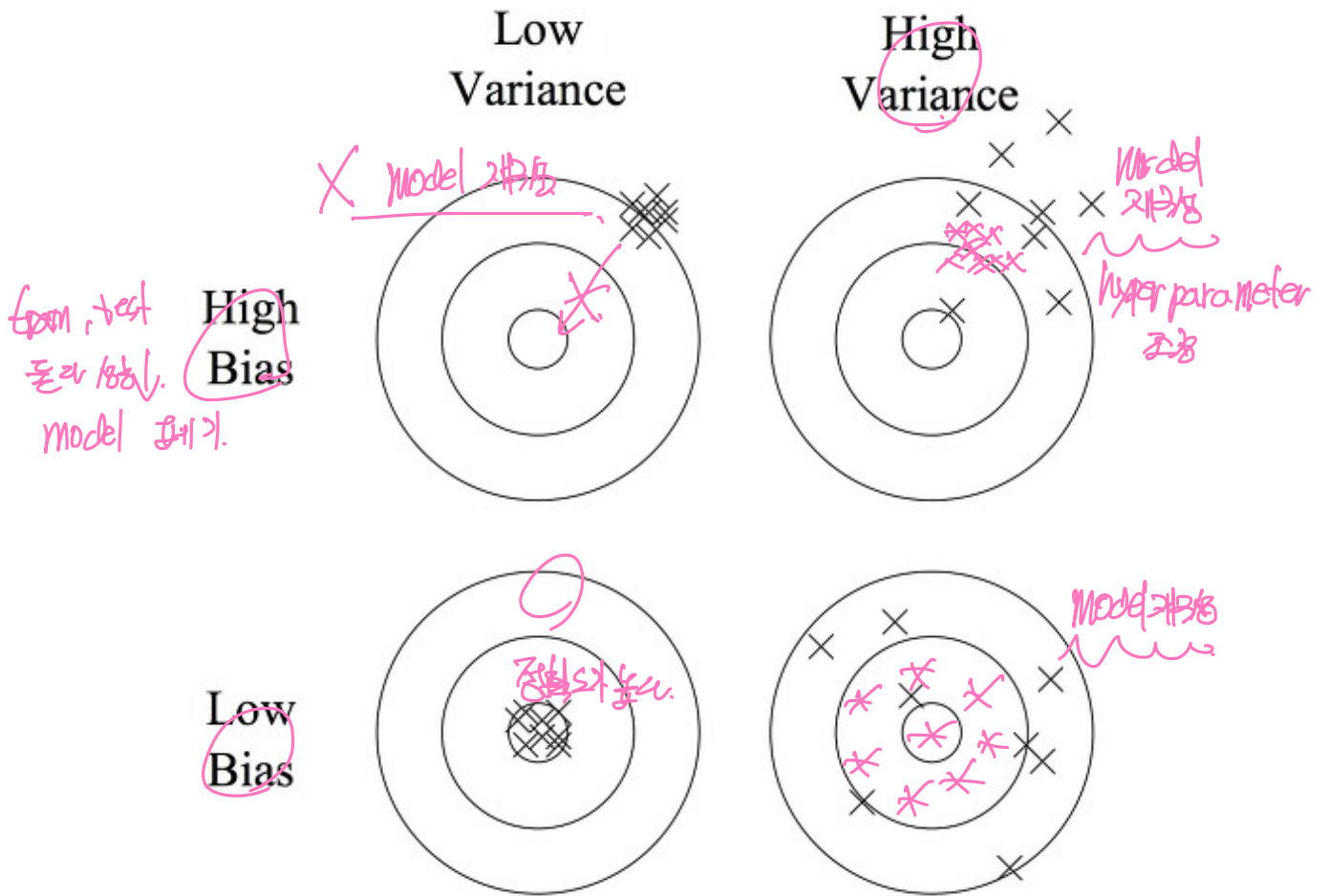
### 1. 첫번째 그림은 과소적합을 나타낸다.

- 훈련 세트와 검증 세트에서 측정한 성능의 간격은 가깝지만 성능 자체가 낮다.
- 과소적합된 모델은 '편향이 크다(high bias)'
- 모델이 충분히 복잡하지 않아 훈련데이터에 있는 패턴을 모두 잡아내지 못한 현상을 나타낸다.
- 모델의 layer 를 넓고 깊게 만들어 Weight 의 갯수를 늘려준다.

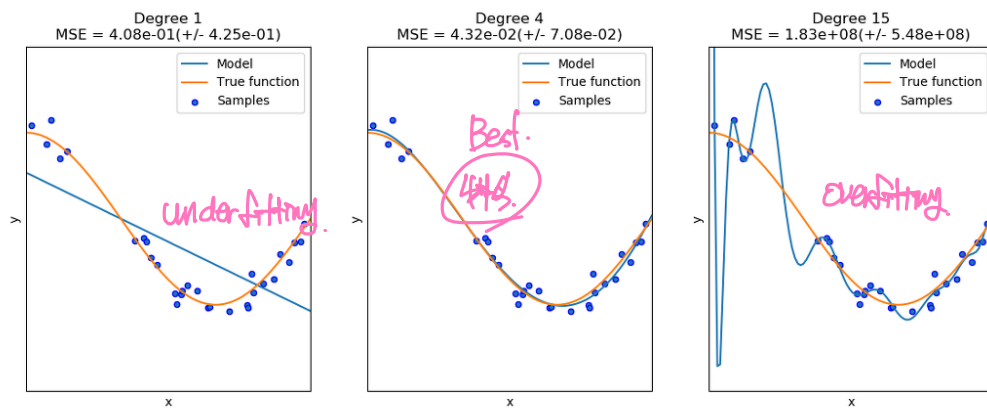
### 2. 두번째 그림은 과대적합의 전형적인 모습을 나타낸다.

- 훈련세트와 검증세트에서 측정한 성능의 간격이 크다.
- 과대적합된 모델을 '분산이 크다.(High Variance)' 라고 한다.
- 훈련 세트에 충분히 다양한 패턴의 샘플이 포함되어 있지 않아 검증세트에 제대로 적응하지 못한것을 의미한다.

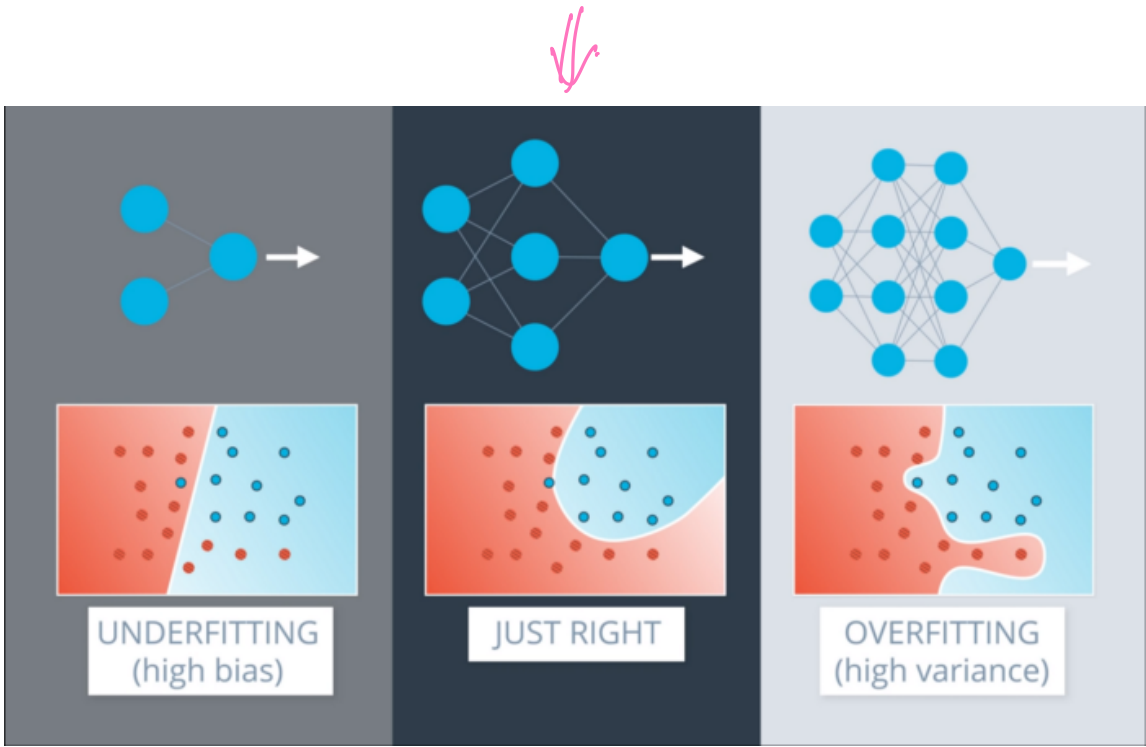
- 많은 훈련 샘플을 모아 검증 세트의 성능을 향상 시킬수 있다.
3. 세번째 그림은 과대적합과 과소적합의 절충된 모습을 나타 내고 있다.



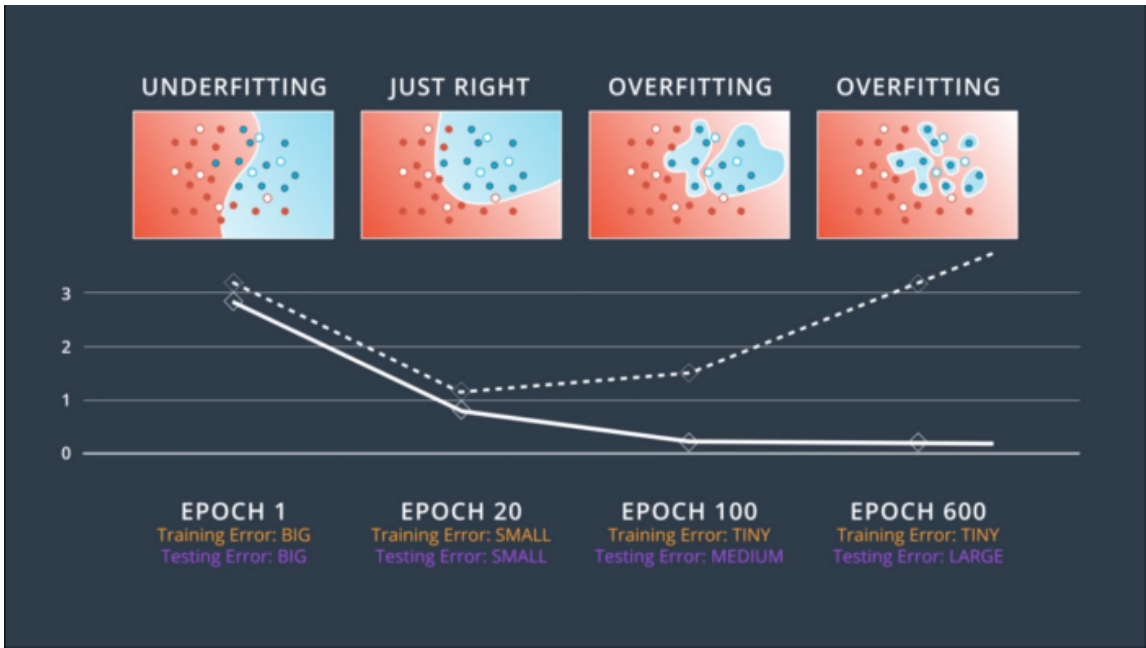
## 모델의 복잡도에 따른 변화



✓  
 $\ln: 0.1, 0.01, 0.001$   
 degree: ~~높게~~

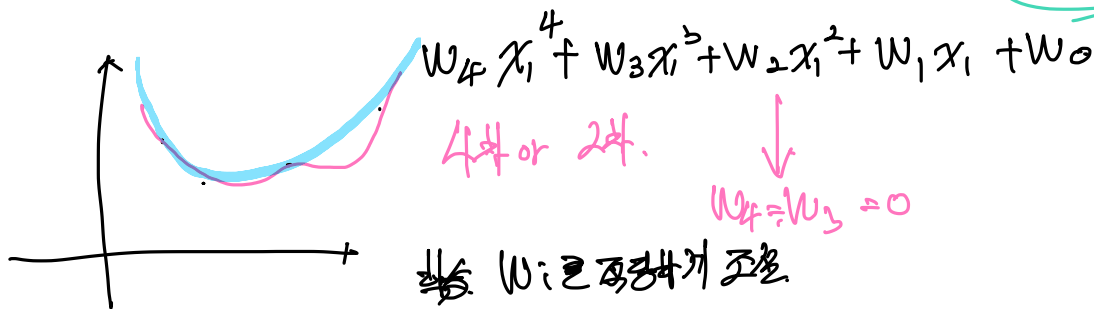
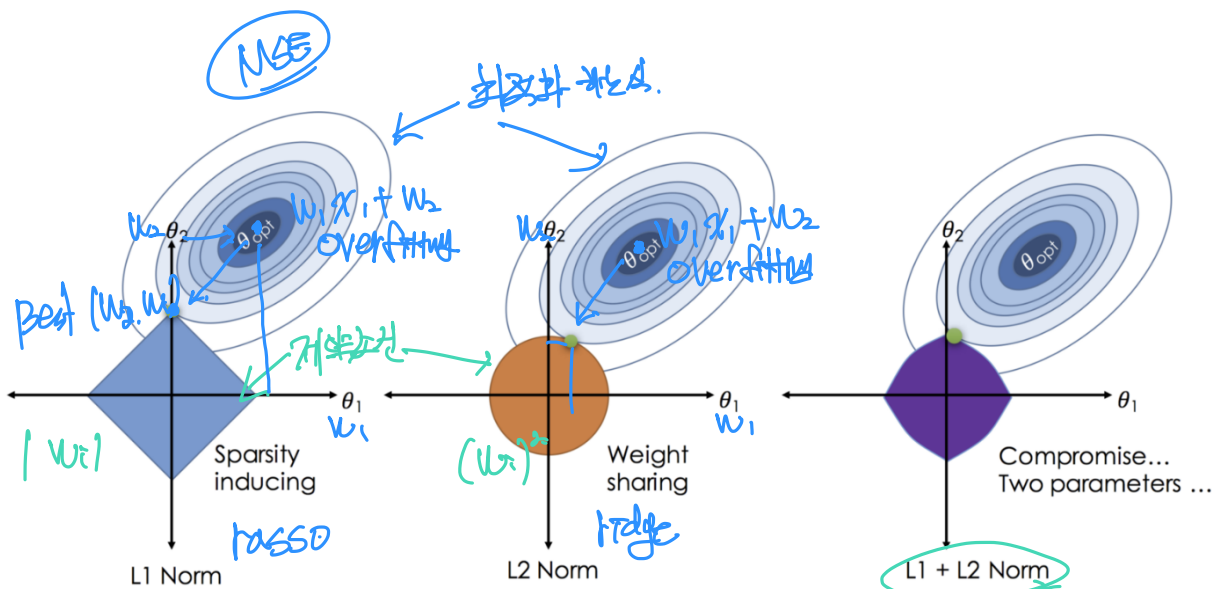


에포크와 손실함수의 그래프로 과대적합과 과소적합 분석



<https://towardsdatascience.com/overfitting-vs-underfitting-ddc80c2fc00d>

Overfitting 해결



정리 (정리)  $\Rightarrow 4x_1^4 + 3x_1^3 + 2x_1^2 + x_1 + 5$   
 $\Rightarrow 0.04x_1^4 + 0.007x_1^3 + 2x_1^2 + x_1 + 5$  (?)

ridge  $\rightarrow$  cost function 이 작게!  
 lasso  $\rightarrow$

$$\sum_{i=1}^n (y_i - (w_1 x_i + w_0))^2 + \lambda \sum_{j=0}^1 w_j^2$$

(cost function.)

$$\sum_{i=1}^n y_i^2 + (w_1 x_i + w_0)^2 - 2 y_i (w_1 x_i + w_0) + \lambda \cdot w_0 + \lambda w_1$$

$$= \sum_{i=1}^n y_i^2 + (w_1 x_i)^2 + 2 \cdot w_1 w_0 x_i - 2 y_i x_i w_1 + 2 y_i w_0 + \lambda w_0 + \lambda w_1$$

$$\Rightarrow (\lambda w_0) + (\lambda w_1)$$

최소화

(1)  $\Rightarrow$  (1)  $\Rightarrow$

$w_0 \approx 0.5$   
 $\lambda = 0.3 \Rightarrow 0.2$