

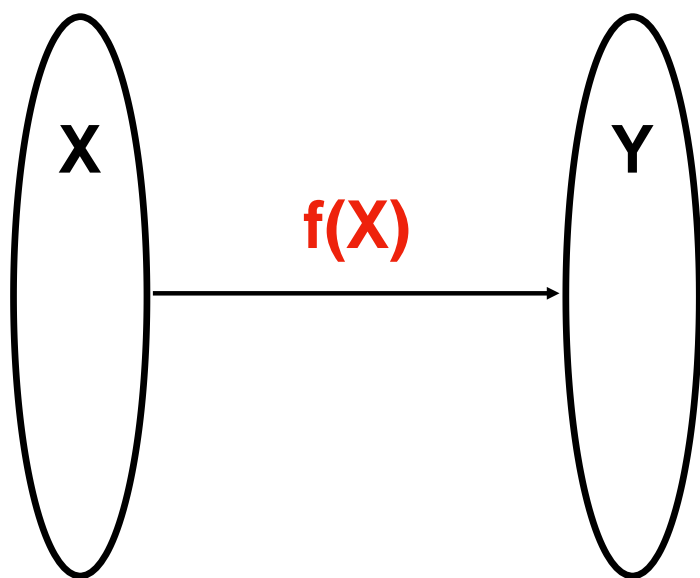
# 딥러닝이란?

딥러닝으로 어떤 문제를 풀 수 있을까?

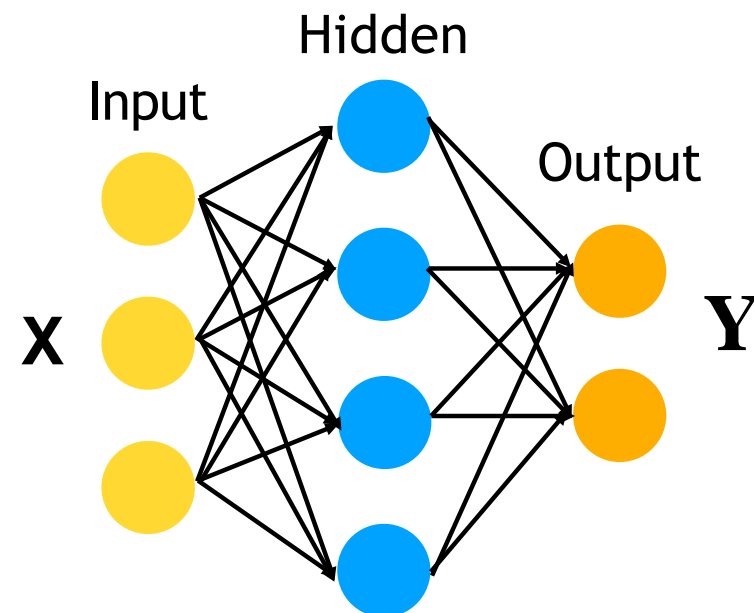
$\hat{Y} = NN(X)$  가  $Y = f(X)$  에 최대한 유사하도록 하는 것.



$$\hat{Y} = NN(X)$$



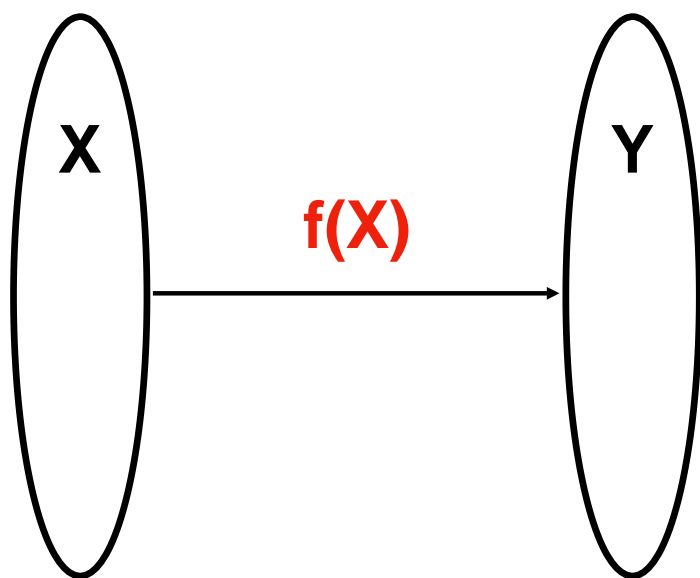
$$f(\mathbf{X}) \approx \hat{f}(\mathbf{X})$$



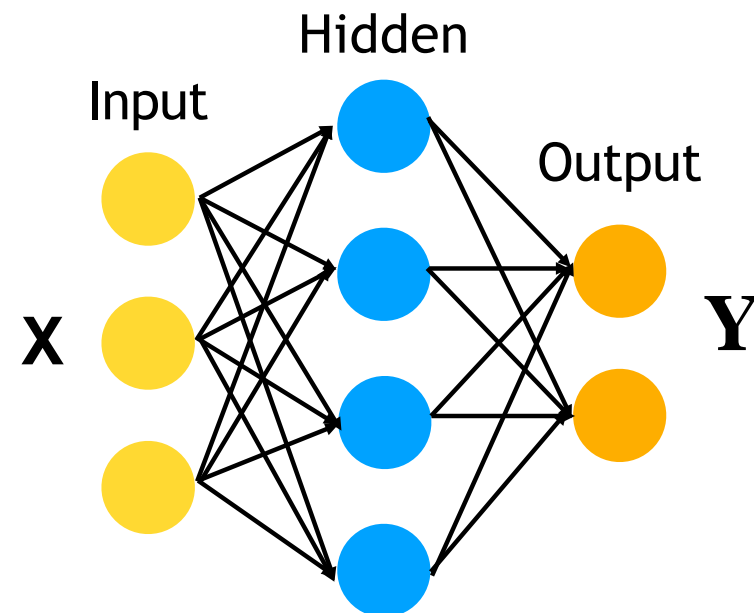
# 딥러닝이란?

딥러닝으로 어떤 문제를 풀 수 있을까?

- 딥러닝은 **Deep Neural Network**을 시키는 것.
- Deep Neural Network은 **하나 이상의** 로 구성된 **Neural Network**이다
- “학습” = 가 에 더더욱 잘 근사하도록 하는 것이다



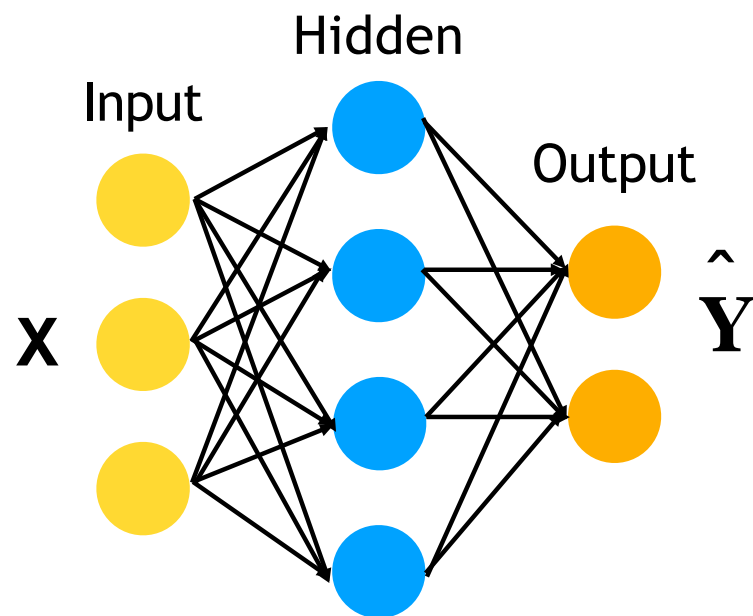
$$f(\mathbf{X}) \approx \hat{f}(\mathbf{X})$$



# Neural Network

기본적인 Neural Network의 구성은:

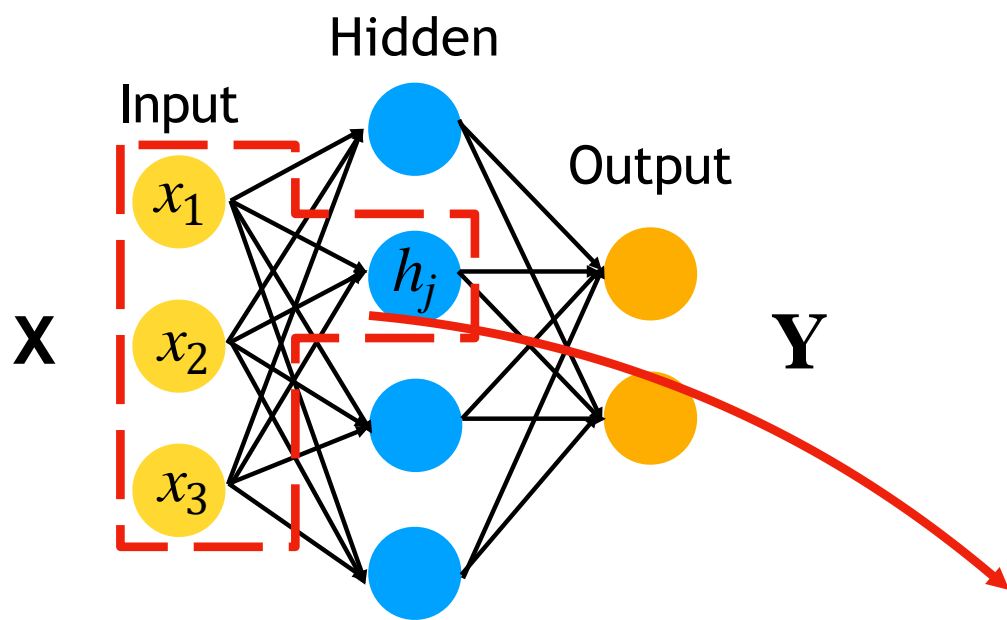
Input Layer + (한 개 이상의)  + Output Layer



- 각 Layer은  들로 구성된다.
- 각 뉴런은 다음 Layer의 뉴런과 연결 (edge) 되어 있다.
- 즉, 뉴런은 이전 Layer의 뉴런들의 출력값을 입력값으로 받는다.
- 각 뉴런은  와  으로 구성된다.

# 딥러닝이란?

## Deep Neural Network란?

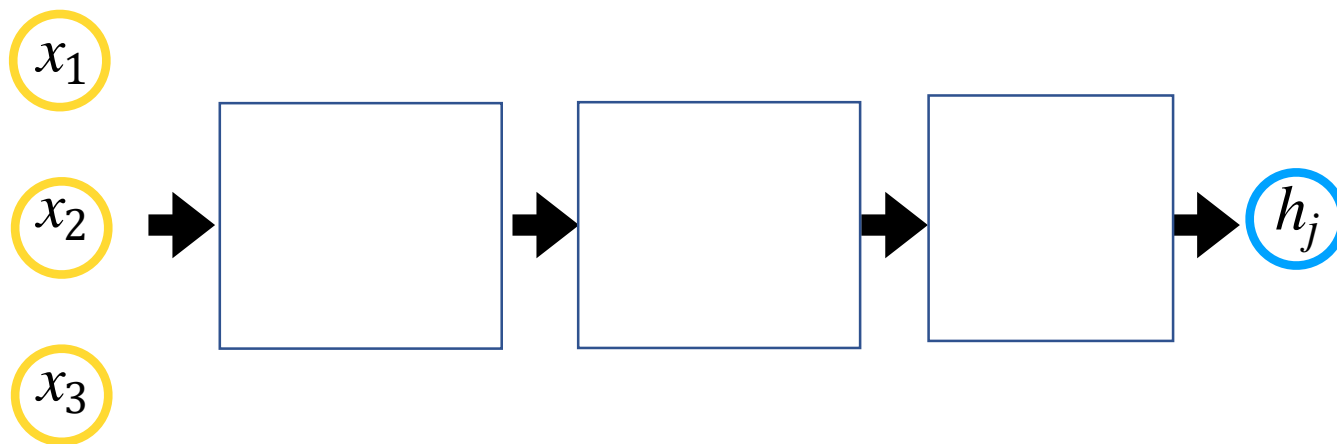


$x_1 \ x_2 \ x_3$

- 뉴럴넷에 입력되는 값
- Input Layer의 1, 2, 3번째 뉴런이 출력하는 값
- 다음 Layer의 Input 값으로 전달

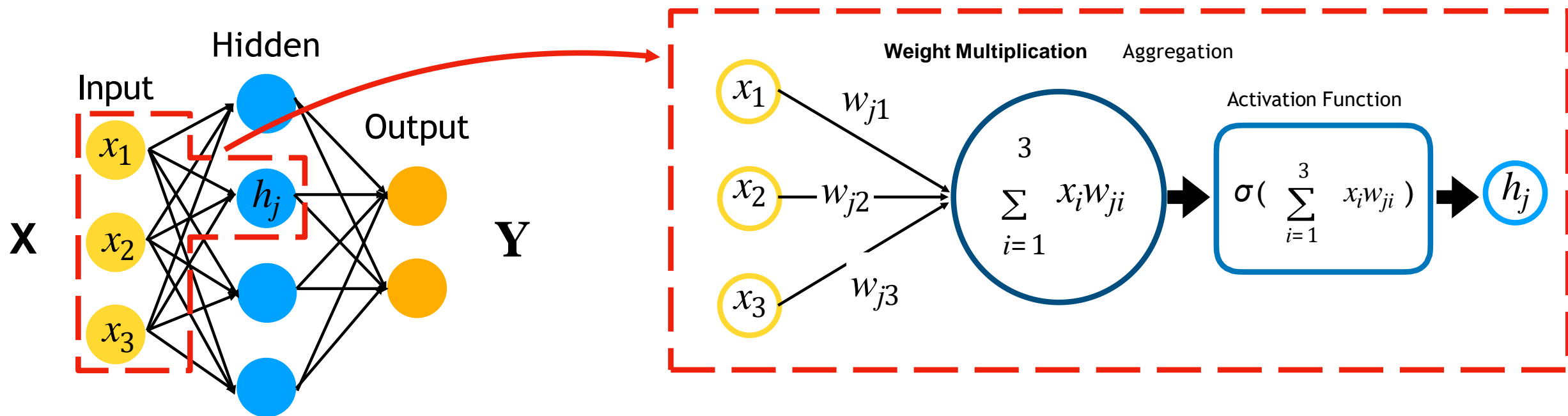
$h_j$

- Hidden Layer의  $j$ 번째 뉴런이 출력하는 값
- 이전 Layer인 Input Layer의 출력값을 입력값으로 사용하여 계산됨
- 다음 Layer의 Input 값으로 전달



# 딥러닝이란?

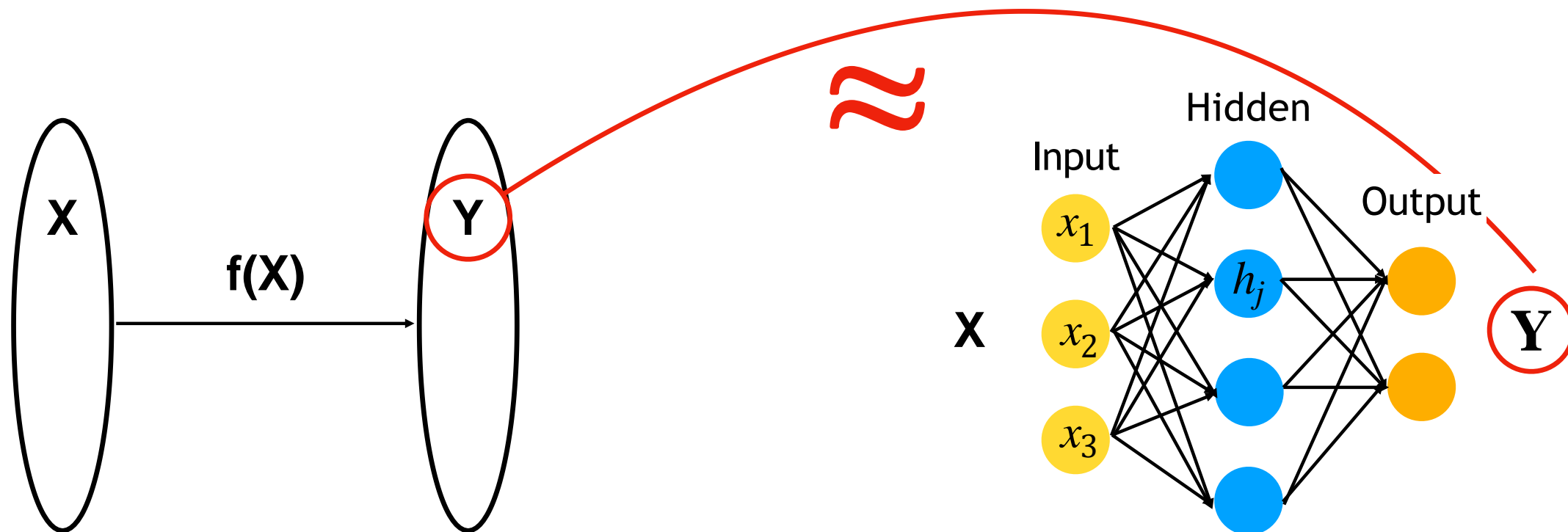
Deep Neural Network란?



# Neural Network의 동작 원리

## Question:

- 어떻게 해야 Neural Network의 출력값인  $\hat{Y}$  가 실제값인  $Y$ 에 최대한 근사  $Y \approx \hat{Y}$  할까?
- 각 Layer의 **weight**  $w$  을 최적화해야 한다.



# Neural Network의 동작 원리

## Question:

- 어떻게 해야 Neural Network의 출력값인  $\hat{Y}$  가 실제값인  $Y$ 에 최대한 근사  $Y \approx \hat{Y}$  할까?
- 각 Layer의 **weight  $w$**  을 **최적화**해야 한다.

- Weight  $w_{ji}$ 이 바뀌면 주어진 Input  $x_i$  값에 대한 Output  $y$  값도 바뀐다
- 주어진 Input  $x_i$ 들에 대해서 최대한 실제값과 유사하게 Output  $\hat{y}$  을 출력해주는 **가중치  $w_{ji}$  들의 조합** 을 찾고 싶은 것
- 각 Layer의 **weight  $w_{ji}$**  을 **적절하게 조정**해서 주어진 Input에 대해서 **출력되는 값  $\hat{y}$  이 실제값** 에 **최대한 잘 근사**하도록 **최적화**하는 것

# Neural Network의 동작 원리

## Weight 값의 최적화

Question:

- Weight 값을 어떻게 정의해야 예측값이 최대한 정확할까?

모름! 처음에는 weight 값을 랜덤하게 정의함!

- Weight 값을 어떻게 최적화해야 모델의 예측값이 더 정확해질 수 있을까?

을 통한 을 최소화하도록  
을 최적화하여 점진적으로 모델의 예측 정확도를 높인다.

\* 즉, 처음에는 랜덤한 weight 값에 따라 모델의 예측값도 random 하지만 weight 값을 최적화하여 점차 모델의 정확도를 높임!



# Deep Learning 기초 개념

## 1. Training (학습 / 훈련) 데이터셋

- 모델을 학습시키는 용도

(딥러닝 모델의 경우) 학습 데이터셋에 대해서  하여  
 가 최소화되도록  최적화

## 2. Validation (검증) 데이터셋

- 모델의  와 Hyperparameter Tuning에 사용

학습 데이터셋에 대한  될 수 있기에 Validation 데이터셋으로 성능 평가

## 3. Test (시험) 데이터셋

- 검증 단계에서 선택한 최적의 모델의  을 평가하는데 사용

Hyperparameter Tuning을 과도하게 적용하는 경우 Validation dataset에 대해서 unintentional overfitting 발생할 수 있다.

### Overfitting란

- 학습 데이터의 Noise마저 학습하여 일반화 성능이 감소하는 것.

### Hyperparameter란

- 모델 구조와 학습 방식에 영향을 주는 Parameter
- 모델의 최종 성능에도 영향을 준다.

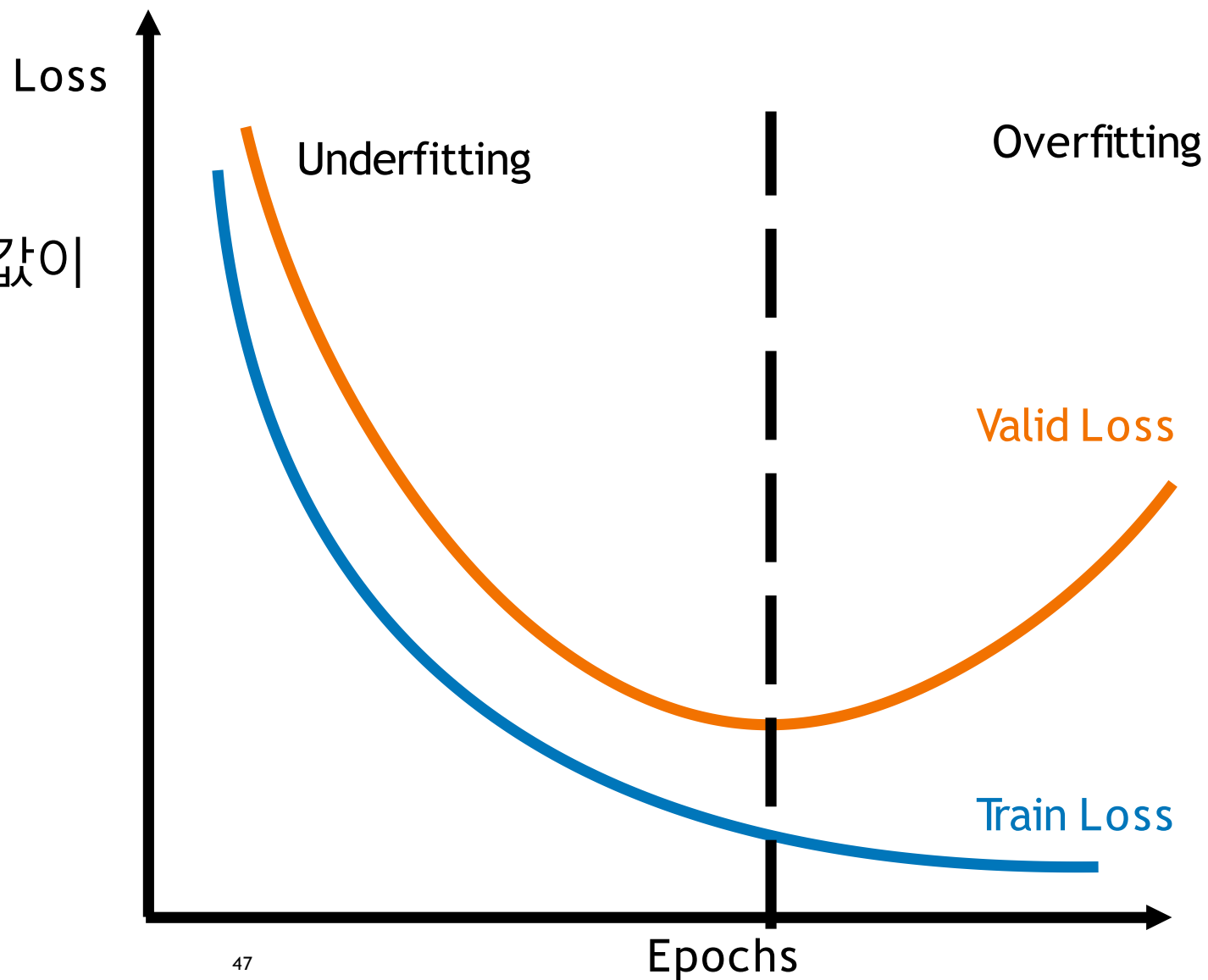
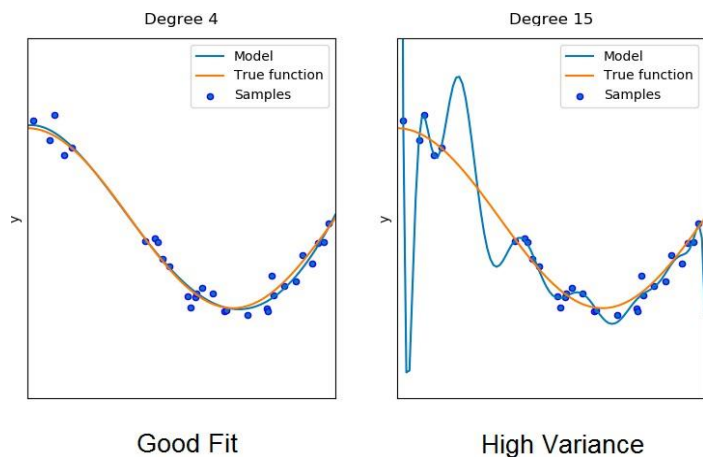
### Hyperparameter Tuning이란

- 가장 최적의 Hyperparameter 조합을 찾는 것.
- Validation 성능이 가장 높은 조합 찾기

# Overfitting

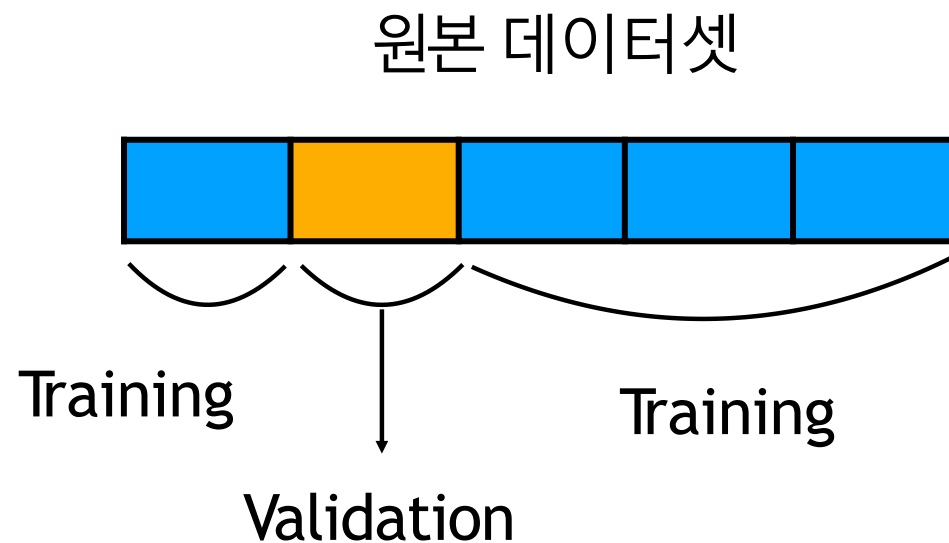
## Overfitting (과적합)이란

- **Unseen data**에 대해서 모델의 예측값이 되지 않는 경우.
- 뉴럴넷 모델이 학습 데이터에 있는 에 대해서도 **학습**하여 가 저하되는 현상.



# K-fold Cross Validation

1. 데이터 분할  
원본 데이터셋을  $K$  개의 서로 겹치지 않는 부분 집합으로 나눈다. (일반적으로  $K=5$  혹은  $10$  사용)
2. 반복 학습 및 검증  
하나의 폴드를 검증 데이터로 사용  
 $K-1$  개의 폴드를 훈련 데이터로 사용하여 모델 학습
3. 성능 측정  
각 반복마다 모델은 검증용 폴드에 대한 성능 평가  
모든  $K$  번의 평가를 완료한 후에 평균 성능 계산
4. 최종 성능 평가 혹은 Hyperparameter Tuning  
Fold들에 대한 평균 성능을 최종 성능 지표로 사용  
혹은 Hyperparameter Tuning에 사용



# Loss와 Evaluation Metric의 차이

- **Loss Function**

모델 학습 단계에서 모델의 가중치를 조정하기 위한 목적으로 예측 오차를 측정  
예측한 값과 실제 타깃 값 사이의 차이

학습 단계에서 Loss를 최소화하는 방향으로 모델의 Weight을 조정  
미분 가능해야한다!

Cross Entropy Loss, Mean Squared Loss, 등등

- **Evaluation Metric**

학습된 모델의 성능을 평가하는데 사용되는 지표  
손실함수와 달리 평가 지표는 더 직관적이다.

정확도 (Accuracy), 정밀도 (Precision), 재현율 (Recall), F1 Score 등등