

Section 2. 딥러닝이란?

목차

- 섹션 0. 강의 소개
- 섹션 1. PyTorch 환경 설정
- 섹션 2. 딥러닝이란?
- 섹션 3. 손실 함수 (Loss Function)
- 섹션 4. 경사 하강 (Gradient Descent)
- 섹션 5. 활성화 함수(Activation Function)
- 섹션 6. 정규화(Regularization), 초기화(Initialization)

Objective

학습 목표

- Deep Learning이란 무엇이고 어디에서 기원하는가?
- 어떤 문제를 풀려고 하는 것인가?
- Neural Network의 기본 구성은 어떻게, 어떻게 학습시키는가?

2-1. 딥러닝이란 뭘까?

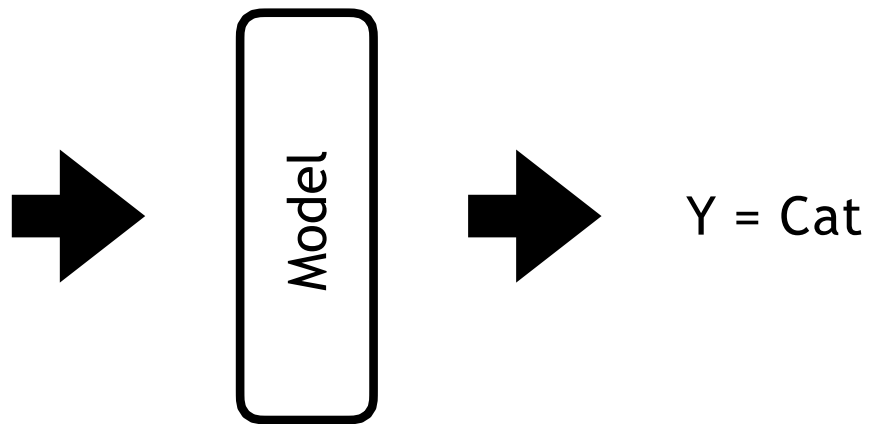
딥러닝이란?

딥러닝으로 어떤 문제를 풀 수 있을까?



Input Data X

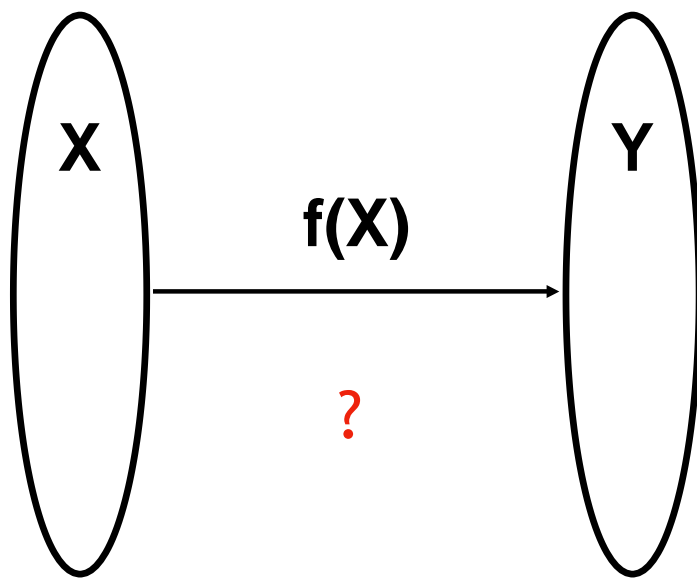
출처: dreamstime.com



딥러닝이란?

딥러닝으로 어떤 문제를 풀 수 있을까?

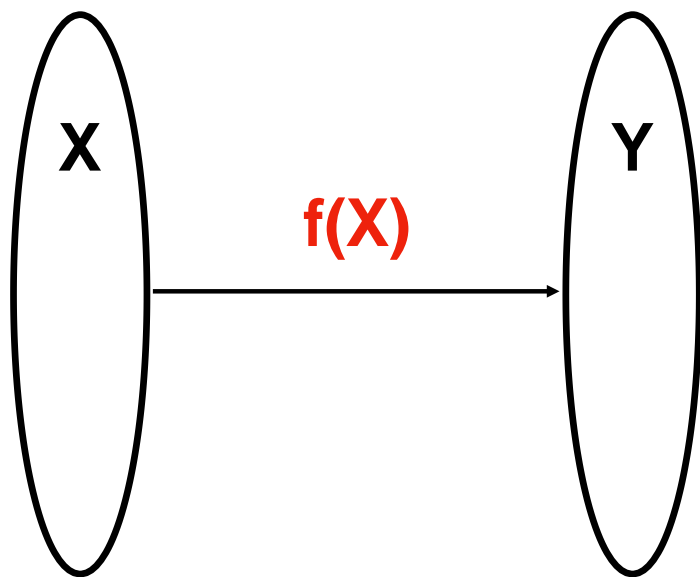
- 입력값 X 가 주어졌을때, Label Y 의 값은 $Y = f(X)$ 을 따른다.
- $X \rightarrow Y$ 로 mapping 해주는 함수 f 을 어떻게 구할까?



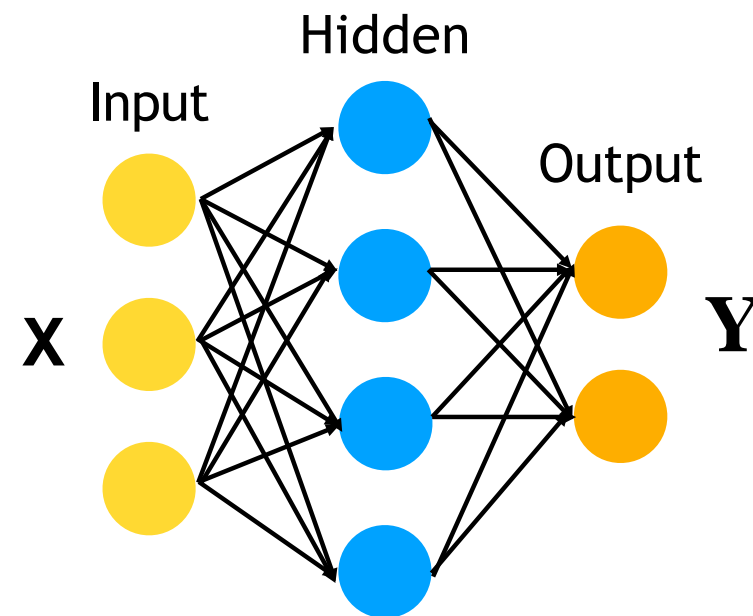
딥러닝이란?

딥러닝으로 어떤 문제를 풀 수 있을까?

- $X \rightarrow Y$ 로 mapping 해주는 함수 $f(\mathbf{X})$ 을 어떻게 구할까?
- 정확한 $f(\mathbf{X})$ 을 구하기 어려울 수 있다.
- $f(\mathbf{X})$ 에 “근사” (approximate) 하는 함수를 Neural Network $\hat{f}(\mathbf{X})$ 으로 모델링 $\hat{Y} = NN(X)$



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

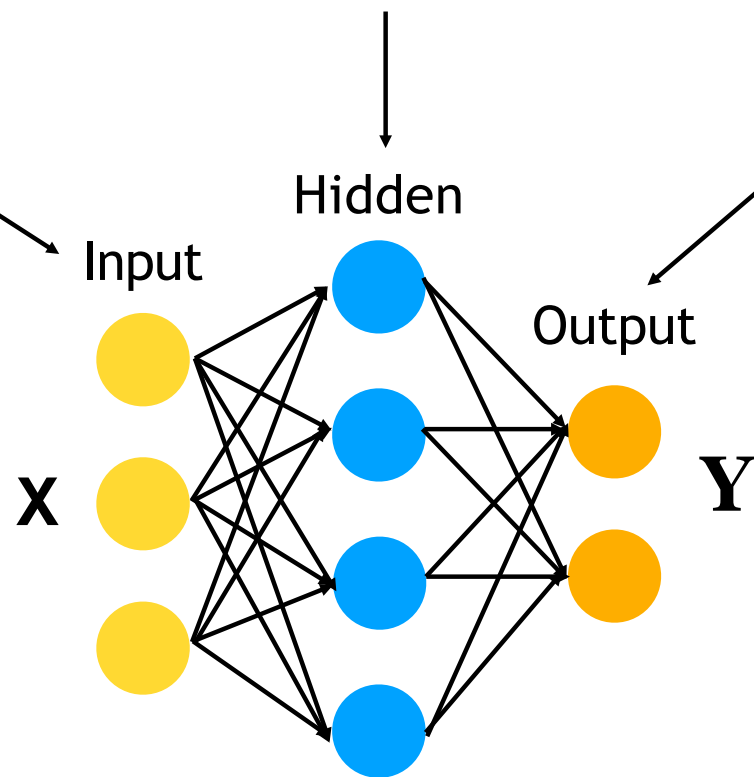


2-2. 뉴럴넷 (Neural Network)은 뭘까?

Neural Network

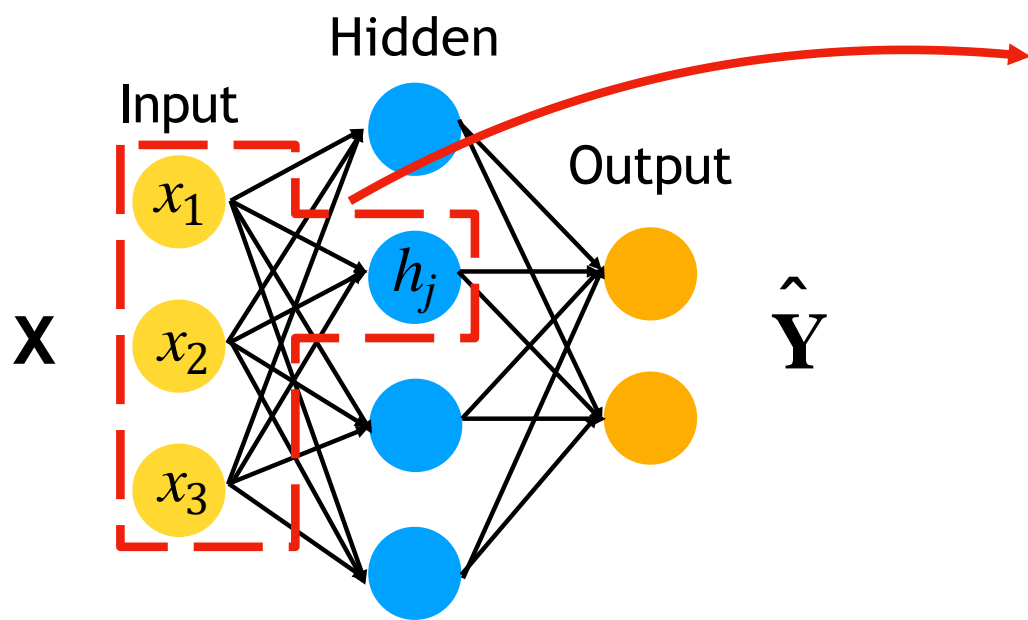
기본적인 Neural Network의 구성은:

Input Layer + (한 개 이상의) Hidden Layer + Output Layer



딥러닝이란?

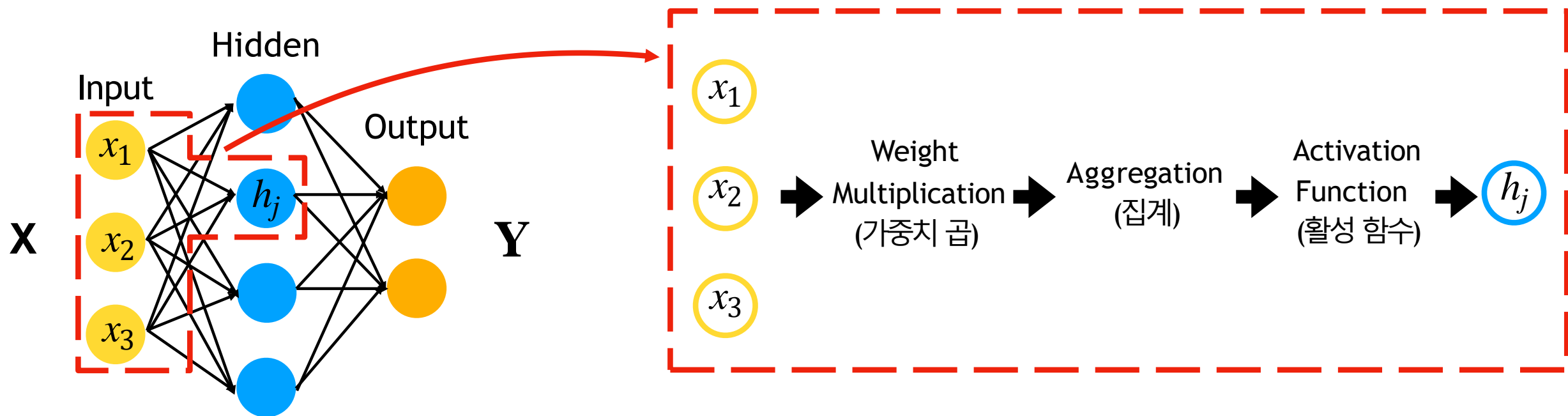
Deep Neural Network란?



- x_1 x_2 x_3
 - 뉴럴넷에 입력되는 값
 - Input Layer의 1, 2, 3번째 뉴론이 출력하는 값
 - 다음 Layer의 Input 값으로 전달
- h_j
 - Hidden Layer의 j번째 뉴론이 출력하는 값
 - 이전 Layer인 Input Layer의 출력값을 입력값으로 사용하여 계산됨
 - 다음 Layer의 Input 값으로 전달

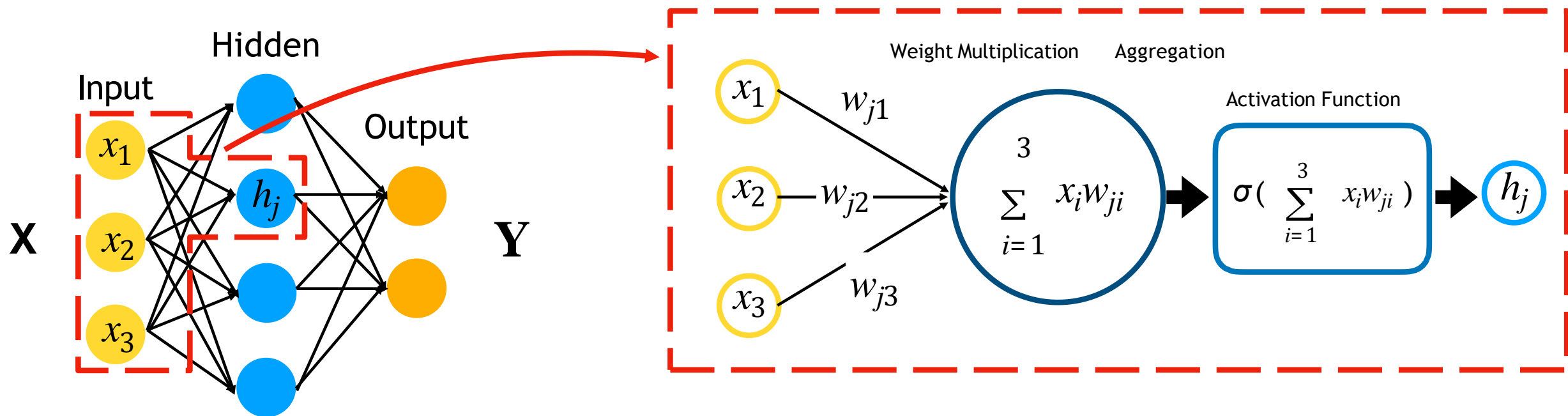
딥러닝이란?

Deep Neural Network란?



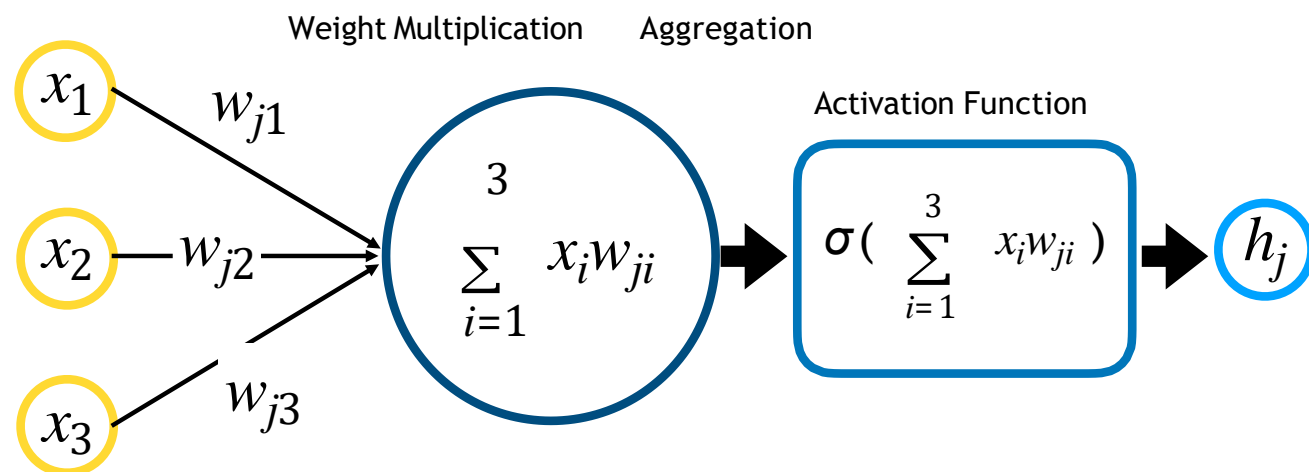
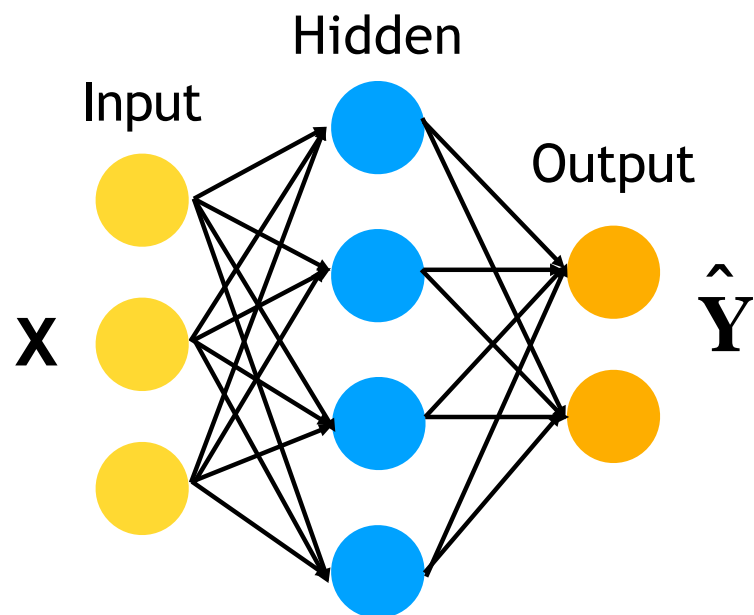
딥러닝이란?

Deep Neural Network란?



Neural Network

- 그렇다면 우리는 왜 이것을 Neural Network라고 부르는 것일까?
- Neural Network가 파생된 기원이 있는걸까?

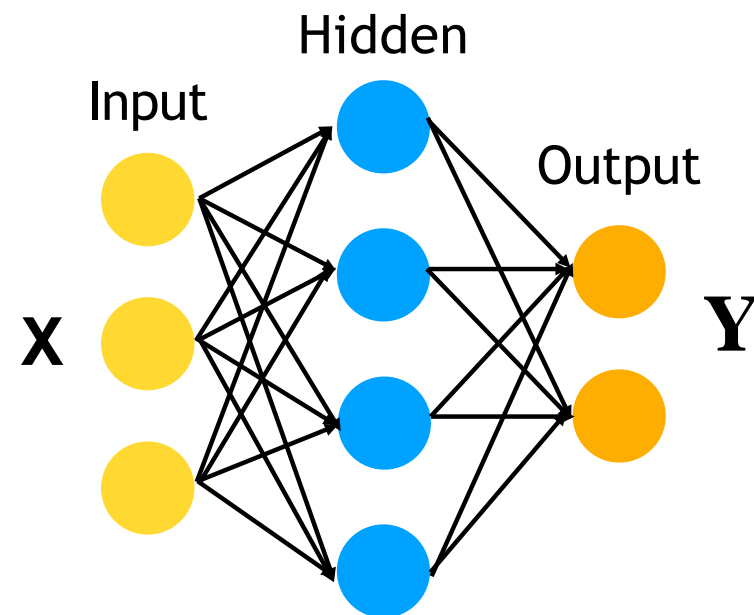


Neural Network의 기원

Neural Network

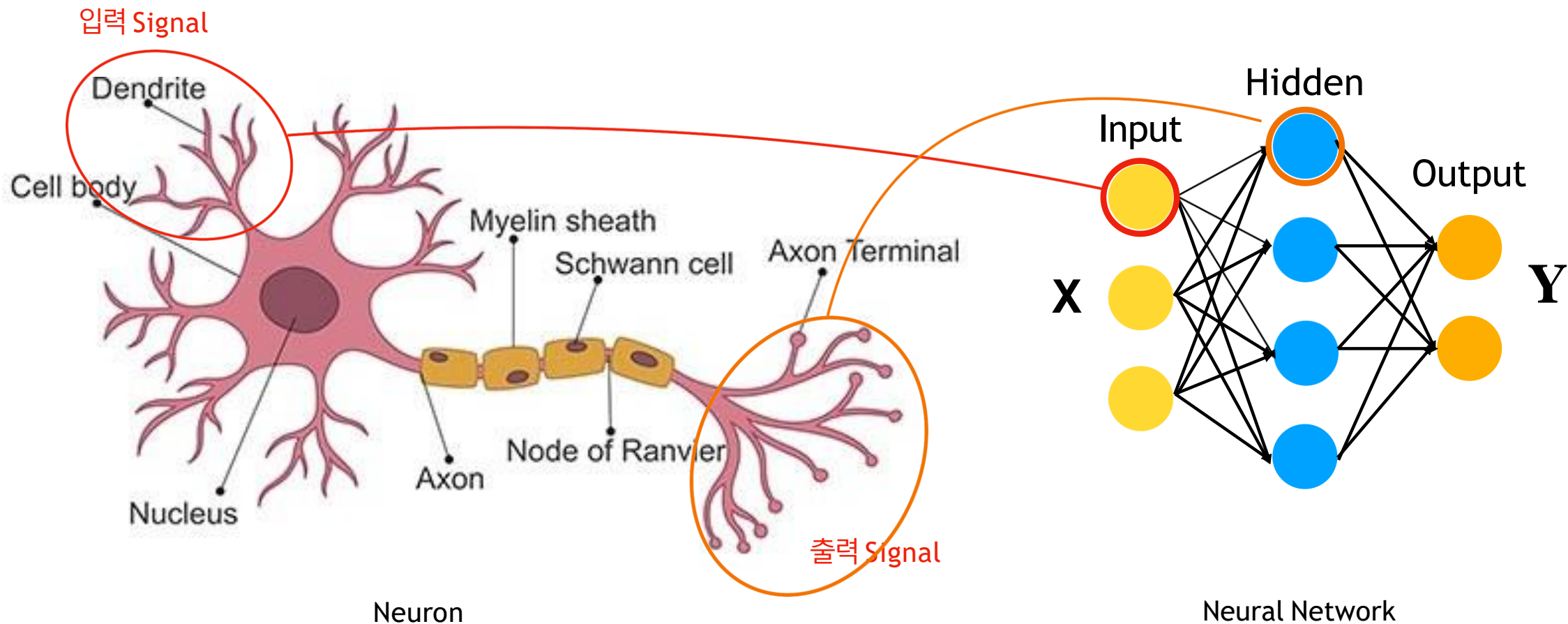
Neural Network의 기원

- 왜 Neural Network라고 부르는 것일까?
- 기원: 뇌의 뉴런 (Neuron)



Neural Network

Neural Network의 기원: 뇌의 Neuron

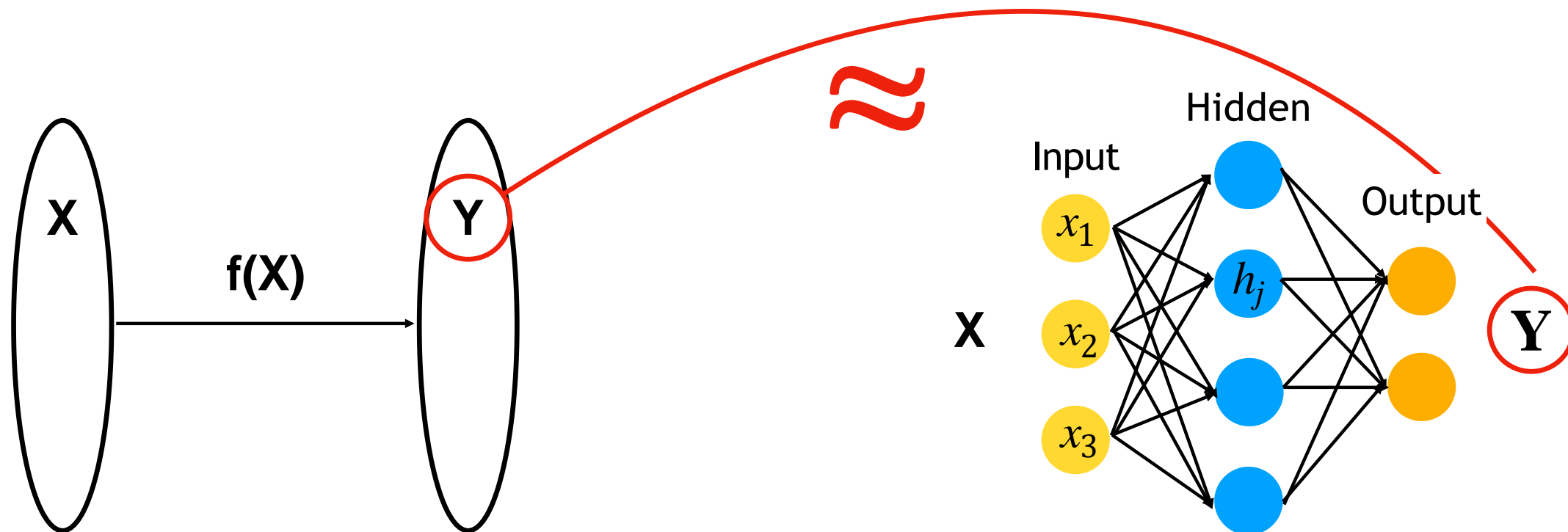


2-3 Neural Network의 동작 원리

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?

Gradient Descent (경사 하강)을 통한 **Loss function (손실 함수)** 값을 최소화하도록

weight 값을 최적화하여 점진적으로 모델의 예측 정확도를 높인다.

2-4. Deep Learning 실무 기초 개념

Deep Learning 실무 기초 개념

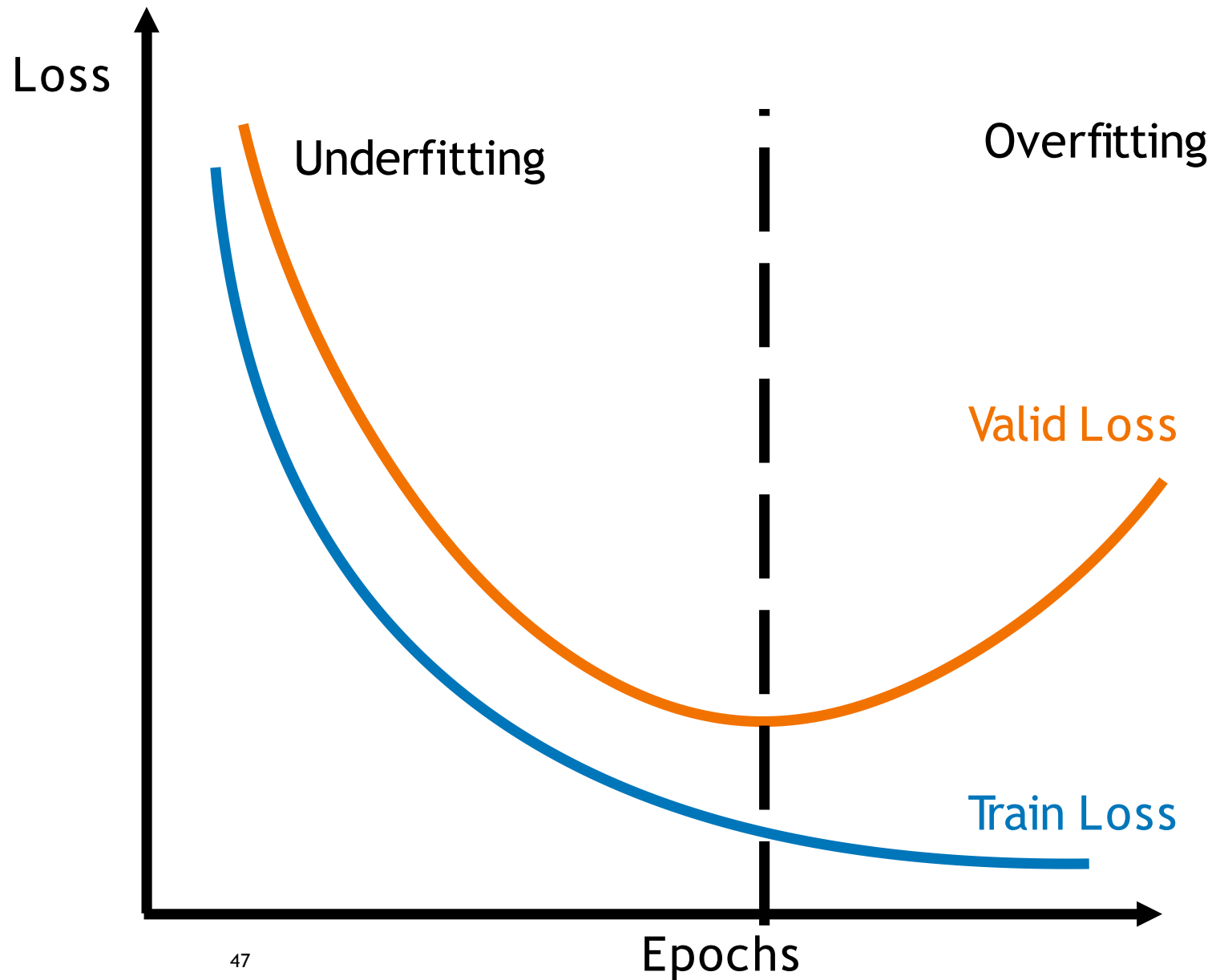
목차:

- Training, Validation, Test dataset
- Overfitting
- K-fold Cross Validation
- Hyperparameter Tuning
- Loss와 Evaluation Metric

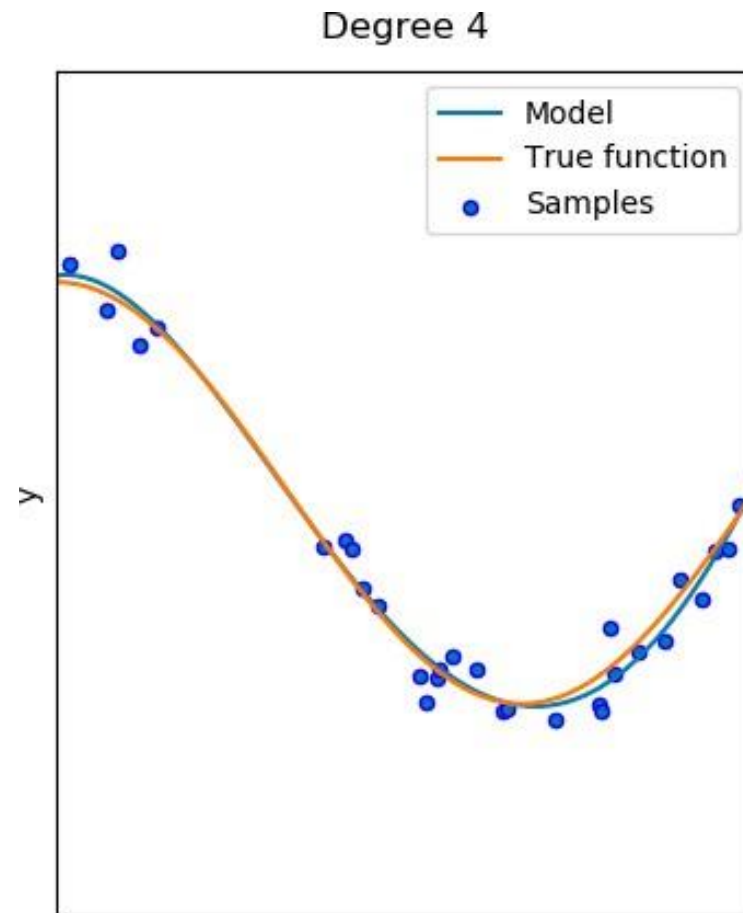
Deep Learning 실무 기초 개념

1. Training (학습 / 훈련) 데이터셋
2. Validation (검증) 데이터셋
3. Test (시험) 데이터셋

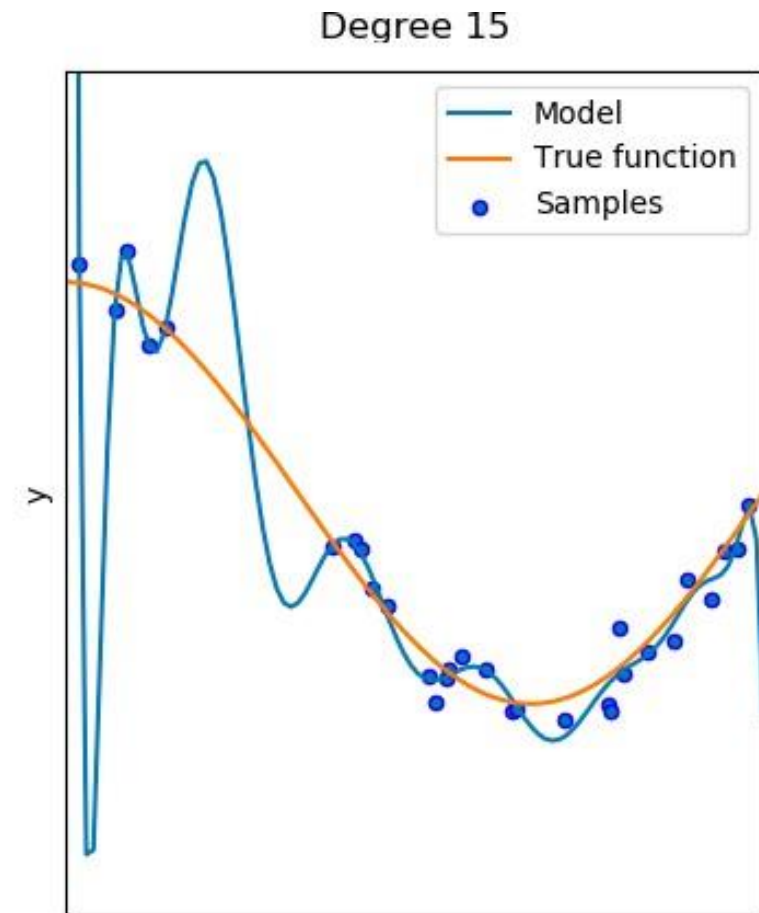
Overfitting



Overfitting



Good Fit



High Variance

Deep Learning 실무 기초 개념

K-fold Cross Validation

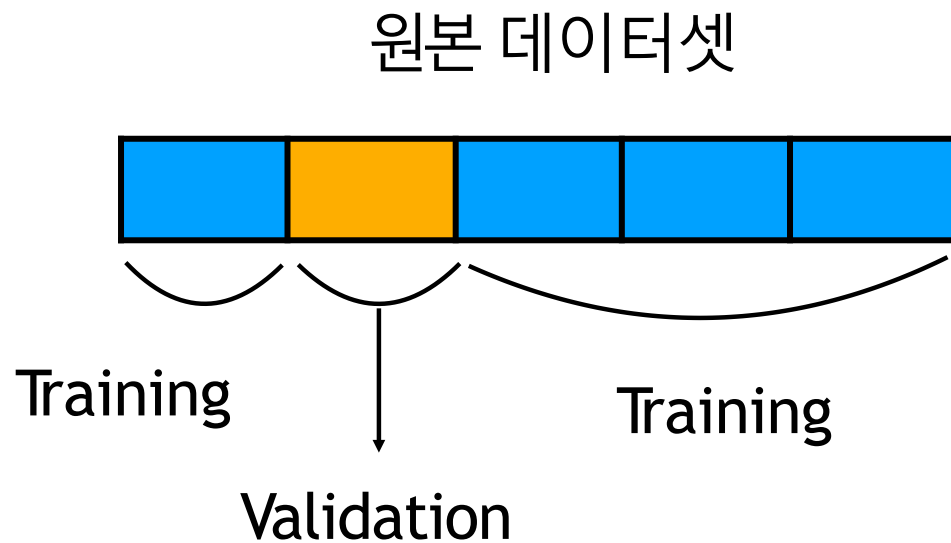
1. 데이터 분할

2. 반복 학습 및 검증

- 하나의 폴드를 검증 데이터로 사용
- K-1개의 폴드를 훈련 데이터로 사용하여 모델 학습

3. 성능 측정

4. 최종 성능 평가 혹은 Hyperparameter Tuning



Deep Learning 실무 기초 개념

K-fold Cross Validation

1. 데이터 분할
2. 반복 학습 및 검증
- 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

Deep Learning 실무 기초 개념

K-fold Cross Validation

1. 데이터 분할
2. 반복 학습 및 검증
3. 성능 측정
4. 최종 성능 평가 혹은 Hyperparameter Tuning
 - Fold들에 대한 평균 성능을 최종 성능 지표로 사용
 - 혹은 Hyperparameter Tuning에 사용

Deep Learning 실무 기초 개념

K-fold Cross Validation

- 장점:
 - 데이터를 효과적으로 활용하여 모델의 성능을 평가
 - 과적합 문제를 예방하고 모델의 일반화 성능을 더 신뢰성 있게 추정
 - 검증 데이터의 선택에 따라 모델의 평가 결과가 크게 변하지 않는다

Deep Learning 실무 기초 개념

Loss와 Evaluation Metric의 차이

- **Loss Function**

- 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**

Deep Learning 실무 기초 개념

Loss와 Evaluation Metric의 차이

- Loss Function
- **Evaluation Metric**
 - 학습된 모델의 성능을 평가하는데 사용되는 지표
 - 손실함수와 달리 평가 지표는 더 직관적이다.
 - 정확도 (Accuracy), 정밀도 (Precision), 재현율 (Recall), F1 Score 등등

Deep Learning 실무 기초 개념

Loss와 Evaluation Metric의 차이

- Loss function:
 - 모델 학습 단계에서 모델의 가중치를 조정하기 위한 목적으로 예측 오차를 측정
- Evaluation Metric:
 - 학습된 모델의 성능을 평가하고 보고하기 위해 사용

2-5. [실습] PyTorch 기초-Tensor

PyTorch의 Tensor

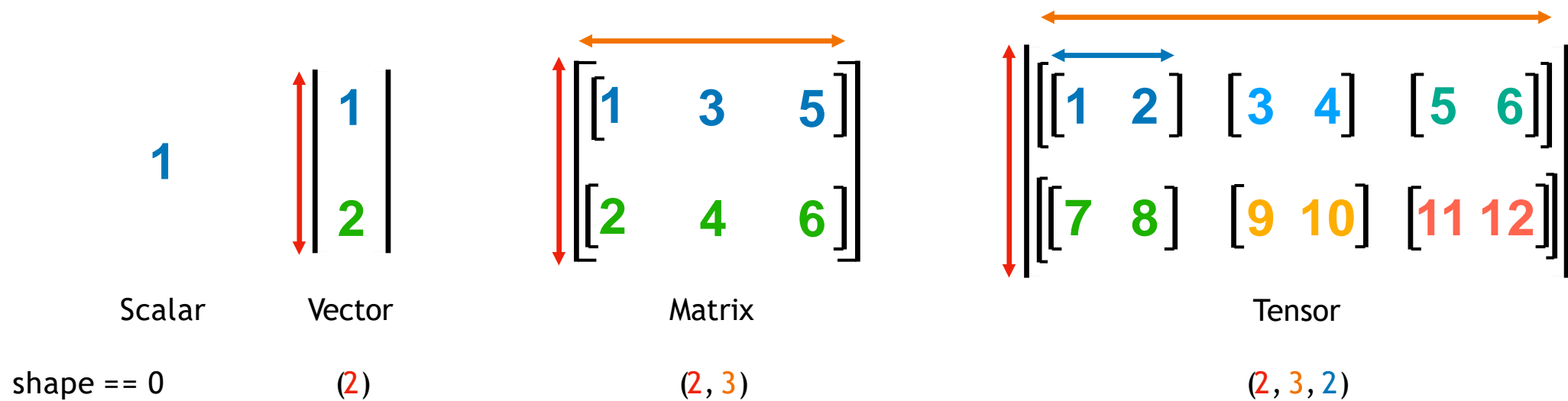
Tensor (torch.Tensor)

- Numpy의 배열 (array)와 행렬 (matrix)와 매우 유사한 자료구조
- PyTorch에서 scalar, vector, matrix, tensor등등을 표현하는데 사용

PyTorch의 Tensor

Tensor (torch.Tensor)

- Numpy의 배열 (array)와 행렬 (matrix)와 매우 유사한 자료구조이다.
- PyTorch에서 scalar, vector, matrix, tensor등등을 표현하는데 사용



PyTorch의 Tensor

Tensor (torch.Tensor)

- Numpy의 배열 (array)와 행렬 (matrix)와 매우 유사한 자료구조이다.
- PyTorch에서 scalar, vector, matrix, tensor등등을 표현하는데 사용

