Deep Learning Week 3

How to evaluate/test model

Hanyang Artificial Intelligence Group

Recall: 모델이란 무엇인가?

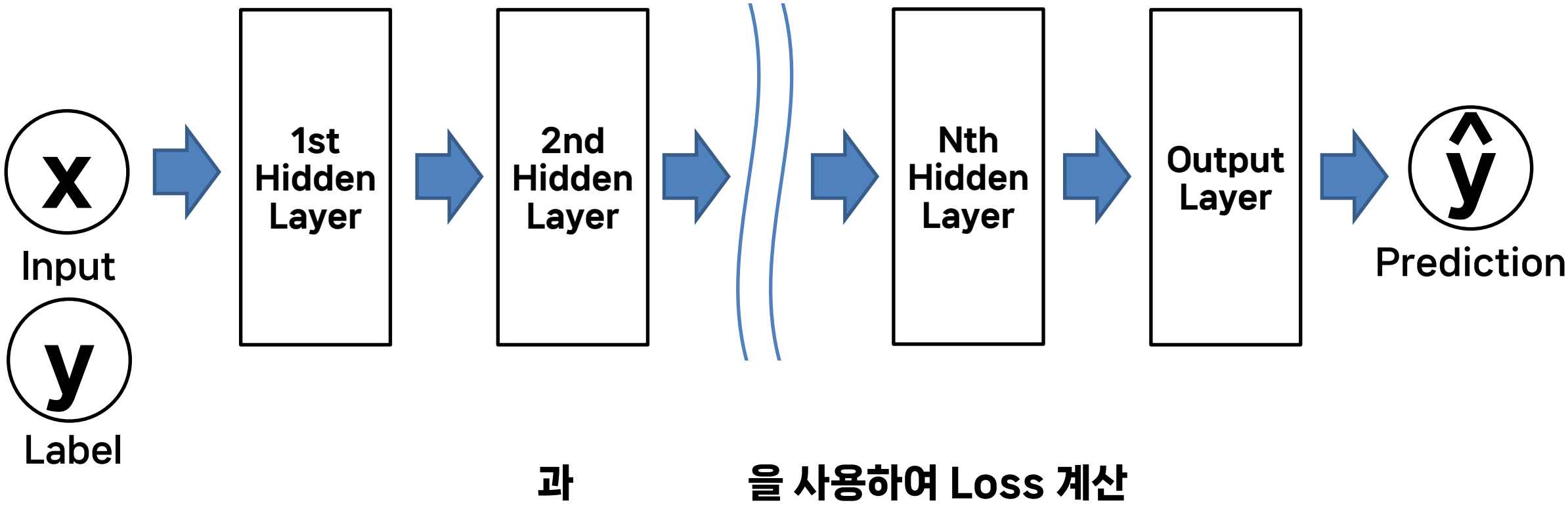
머신러닝 모델 == 함수

- 함수(Function)란? 주어진 input domain의 데이터와 output range의 연관관계
- 딥 러닝을 통해 현실에 존재하는 정답 함수를 최대한 근사하는 모델을 만드는 것이 목표!

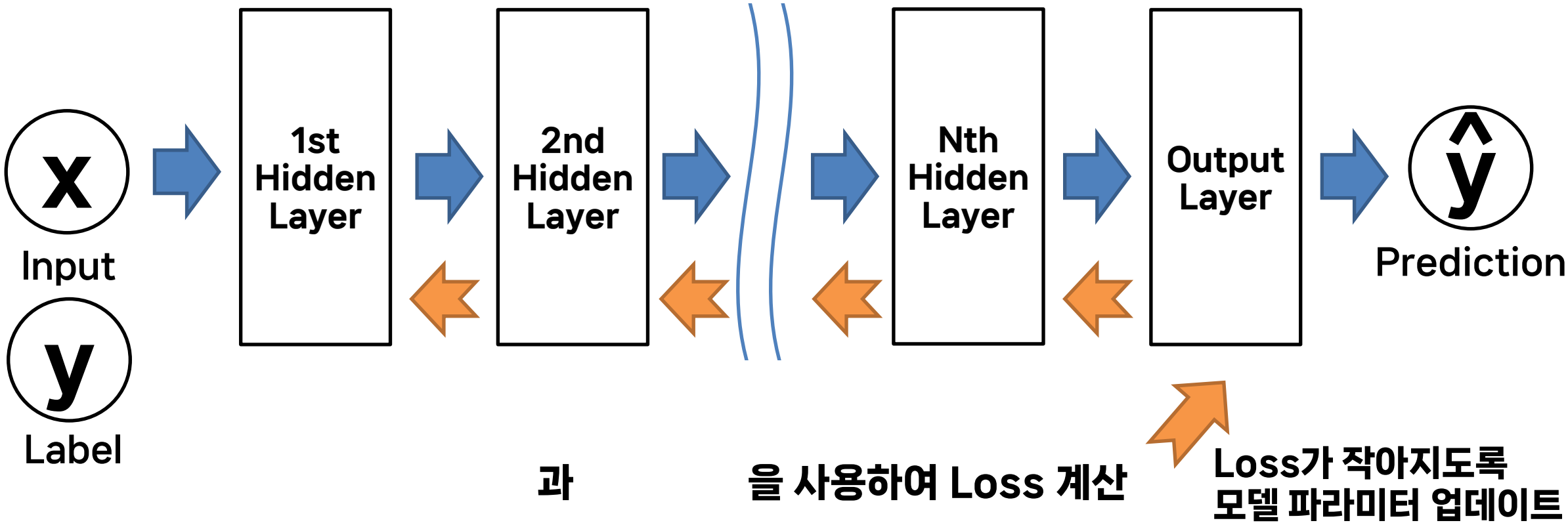
모델 학습의 의미

- 모델 학습이란? 주어진 데이터들의 특성과 패턴을 분석하여 모델이 목적을 더 잘 달성하도록 모델의 파라미터를 업데이트하는 과정
- 주어진 학습 데이터에 대한 task를 얼마나 잘 수행하는지 확인하기 위한 목적 함수(손실 함수)를 정의하여, 학습 과정에서 이 값이 작아지도록 진행
- 목적 함수를 지칭하는 다양한 용어들: objective function, **loss** function, cost function, ...

딥 러닝을 구성하는 Neural Network 모델



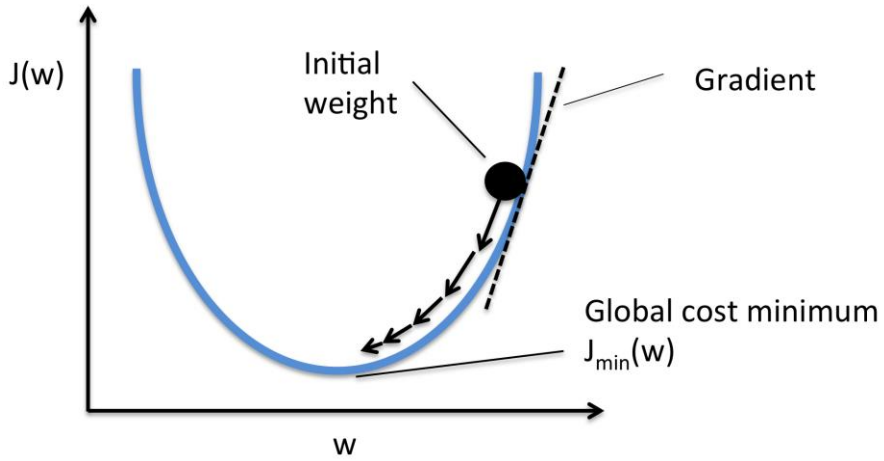
딥 러닝 모델의 back propagation



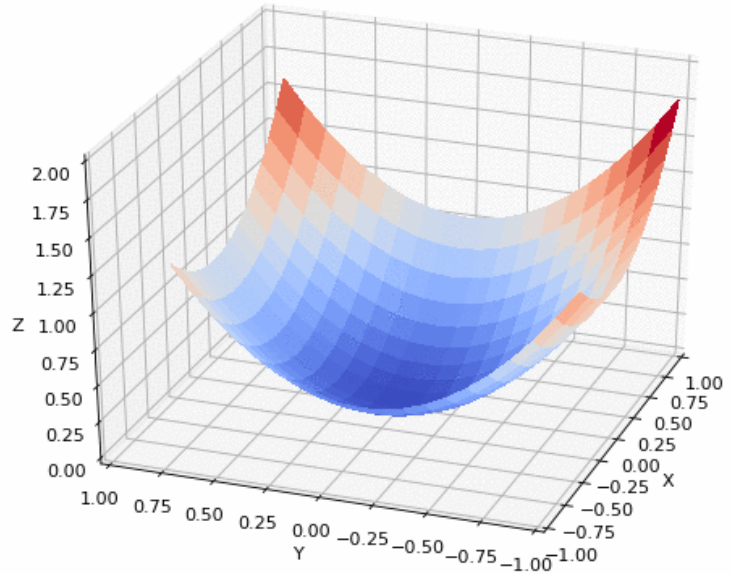
모델 파라미터 업데이트: Optimization

Loss를 최소화하는 최적의 파라미터를 찾자!

- Optimization: 주어진 데이터와 모델을 활용하여 loss값을 계산한 후, loss값이 작아지도록 하는 최적의 파라미터를 찾는 과정
- Gradient descent: loss를 미분하여 감소하는 방향을 찾고, 해당 방향으로 파라미터를 일정 간격만큼 움직이며 최적의 파라미터를 찾는 optimization algorithm의 한 종류
- Learning rate: gradient descent 과정에서 loss가 작아지는 방향으로 파라미터를 얼마나 많이 변화시킬지 결정하는 파라미터. 너무 작거나 크면 학습이 잘 진행되지 않을 수 있음



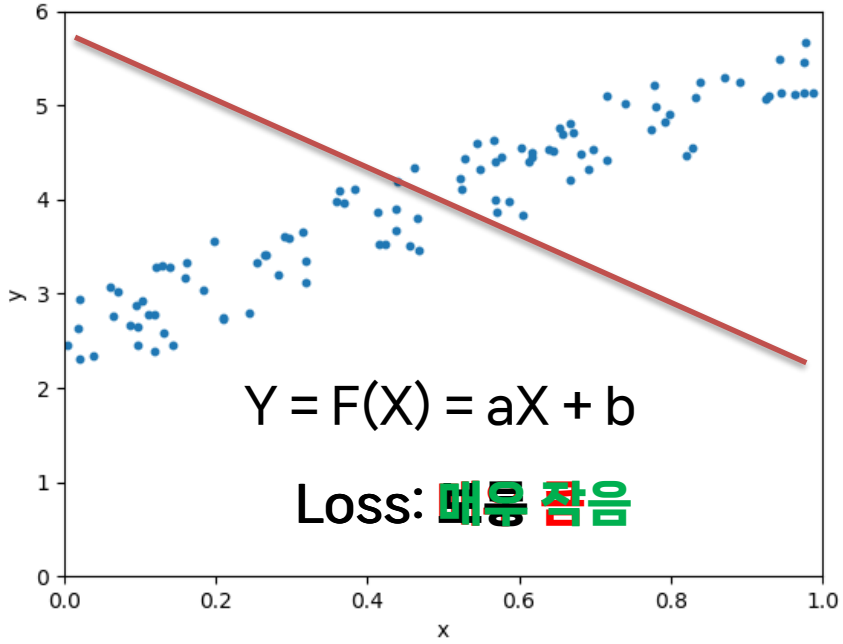
$$\text{repeat until convergence } \{$$
$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$
$$(\text{for } j = 1 \text{ and } j = 0)$$
$$\}$$



머신러닝 모델 예시: Linear Regression

Linear Regression이란?

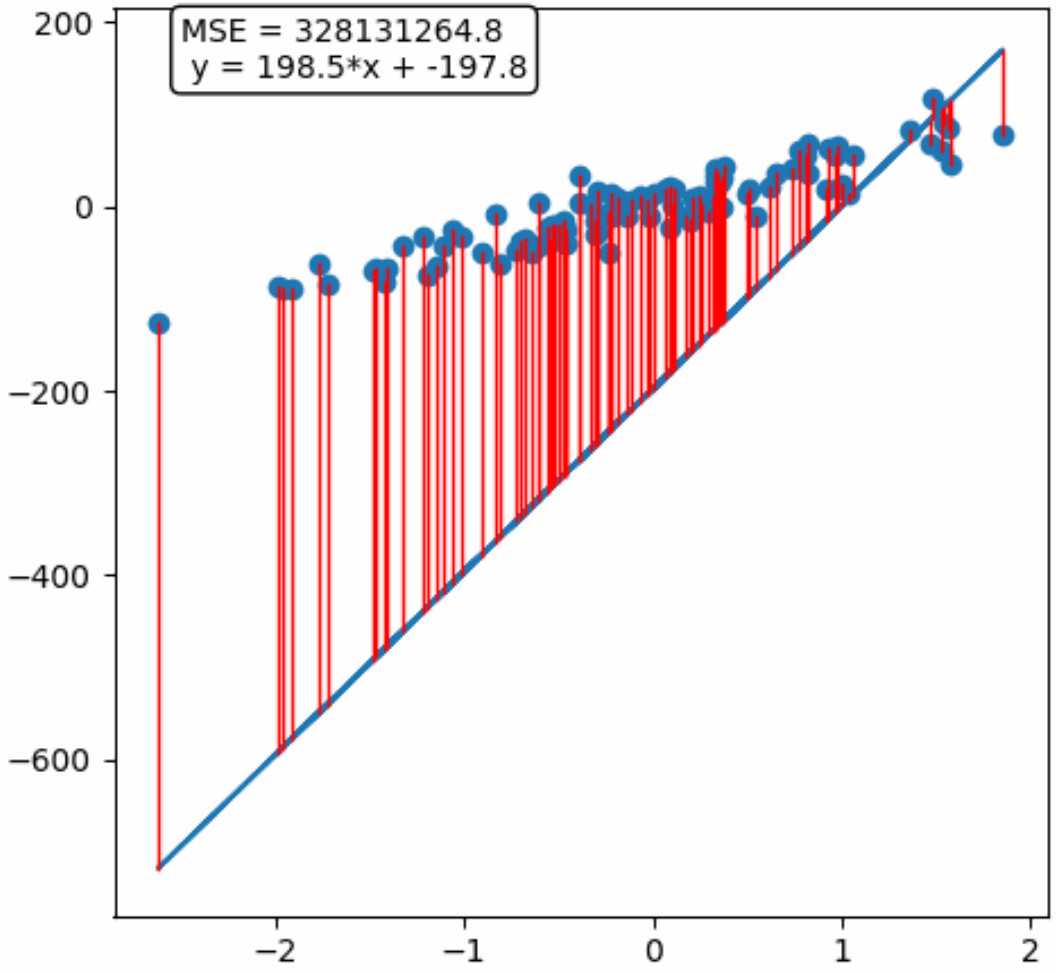
- 주어진 데이터의 분포를 가장 잘 설명할 수 있는 선형 모델(== Linear equation)을 정의하는 것이 목표!
- 학습을 진행하며 모델은 학습 데이터를 더 잘 나타낼 수 있는 형태로 업데이트됨
- Linear Regression의 loss는 각 데이터와 모델 사이의 거리의 평균
- 평균 거리가 작을수록, 모델이 주어진 데이터를 더 잘 표현한다!

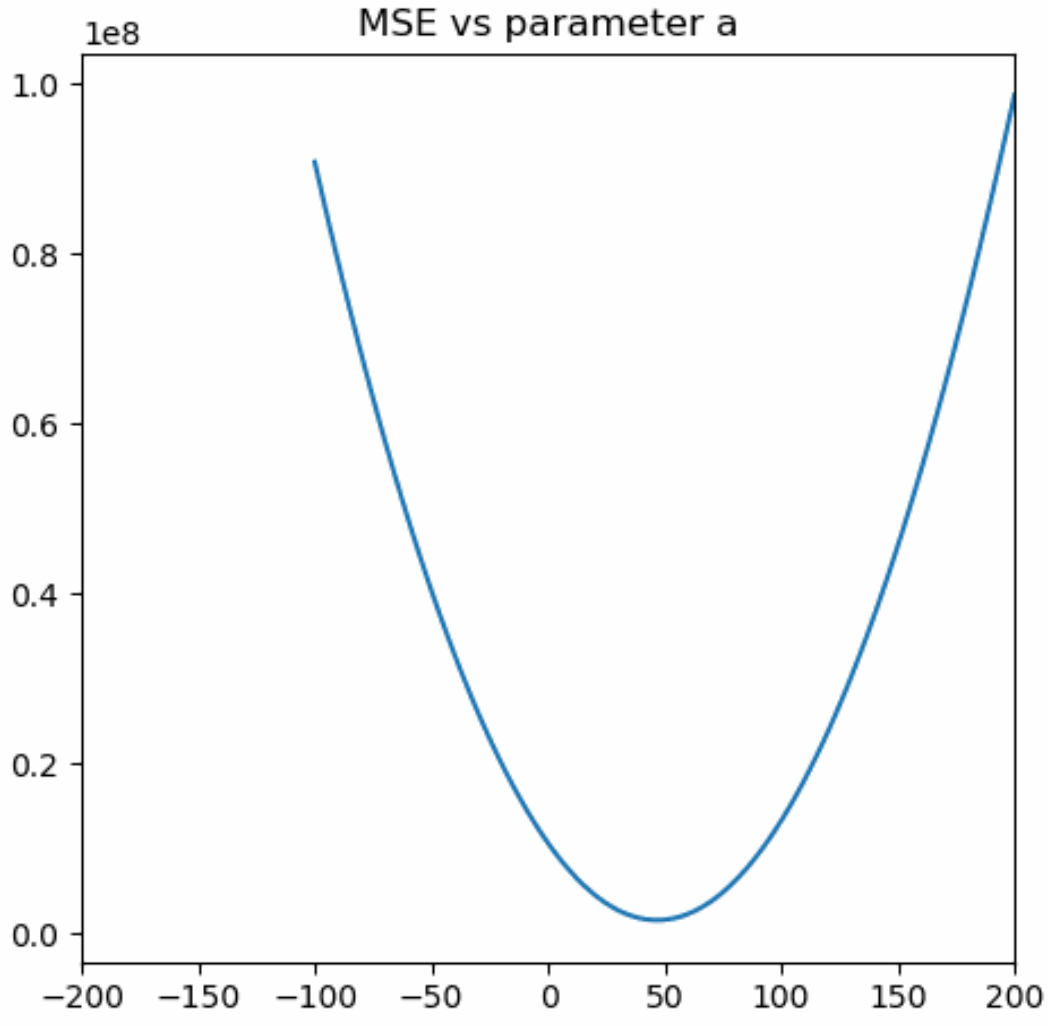


모델 F 학습 과정

- 초기 상태는 랜덤하게 지정 -> 매우 높은 loss
- 학습을 거쳐가며 loss 감소
- 더 이상 loss를 줄이기 어려울 때까지 학습
- 학습 과정에서 모델의 파라미터(a, b)만 업데이트

Linear regression 모델의 gradient descent 과정

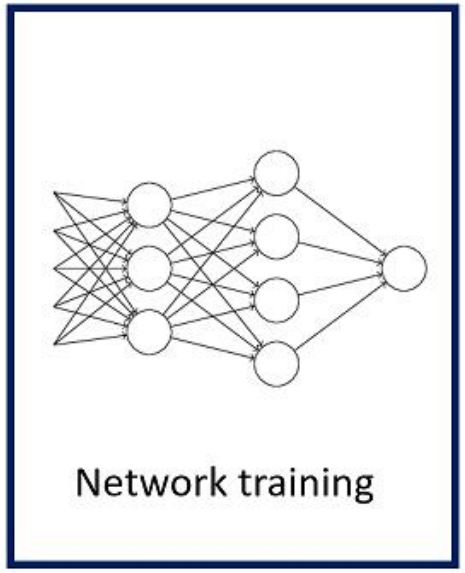




MNIST 데이터셋 분류 문제

- MNIST 데이터셋은 28x28 사이즈의 손글씨 데이터셋
- 0~9까지 총 10개의 클래스를 가지고 있으며, 색상 채널이 없는 흑백 이미지

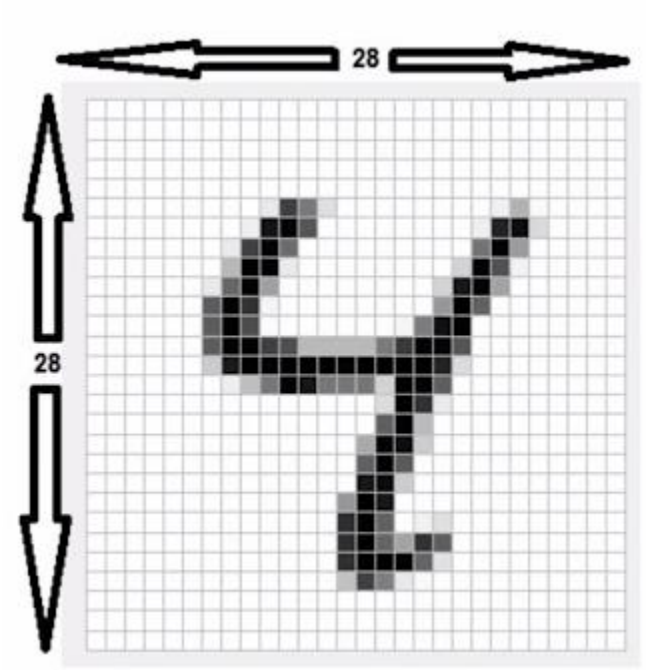
Data & Labels

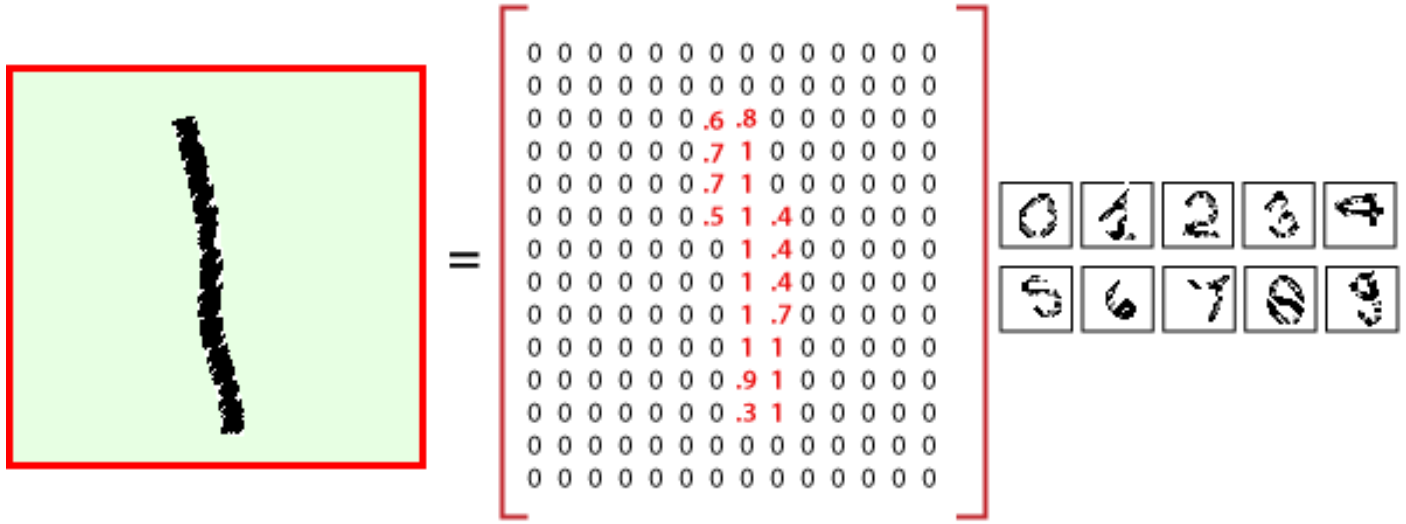


0
1
2
3
4
5
6
7
8
9

MNIST 데이터셋 예시

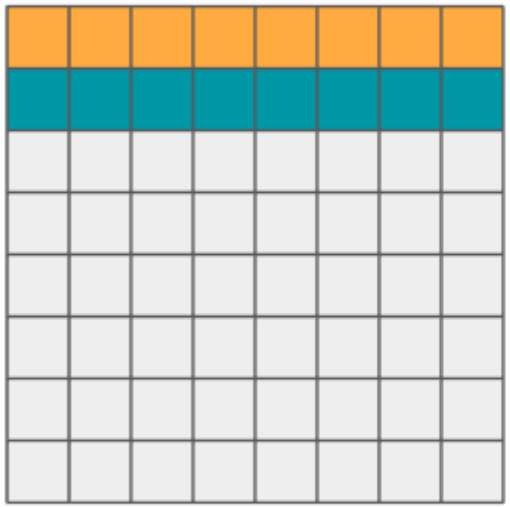
- 각각의 데이터는 28x28 사이즈의 픽셀 값으로 이루어져 있음
- 딥 러닝 모델에 입력할 수 있도록 데이터 전처리가 필요!





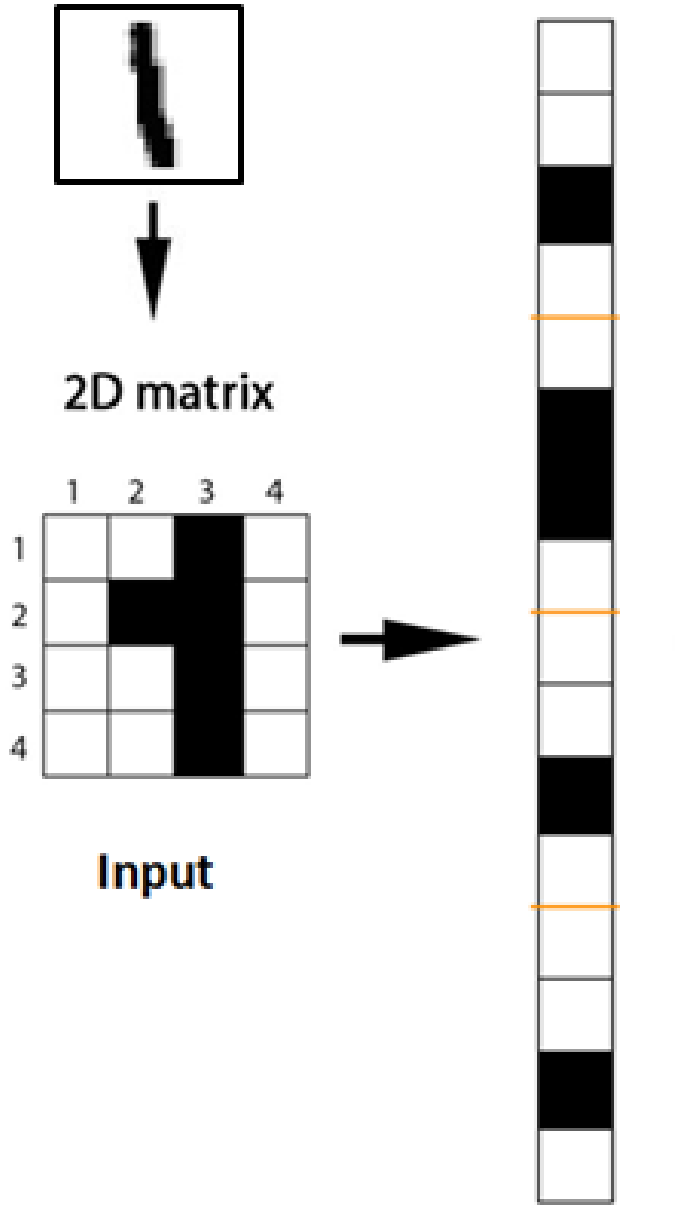
데이터 전처리 과정

- 2차원 행렬 데이터를 1차원 벡터로 변환
- 흔히 flatten 한다고 표현해요!

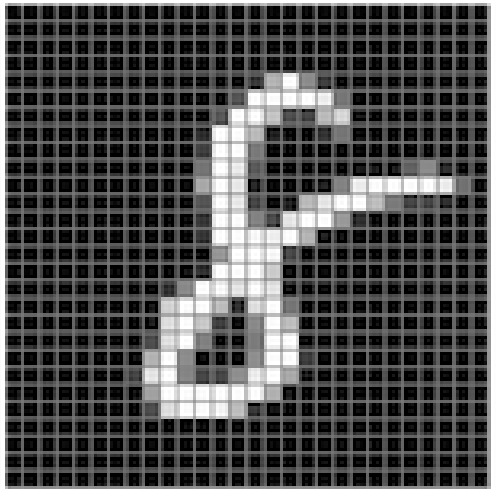


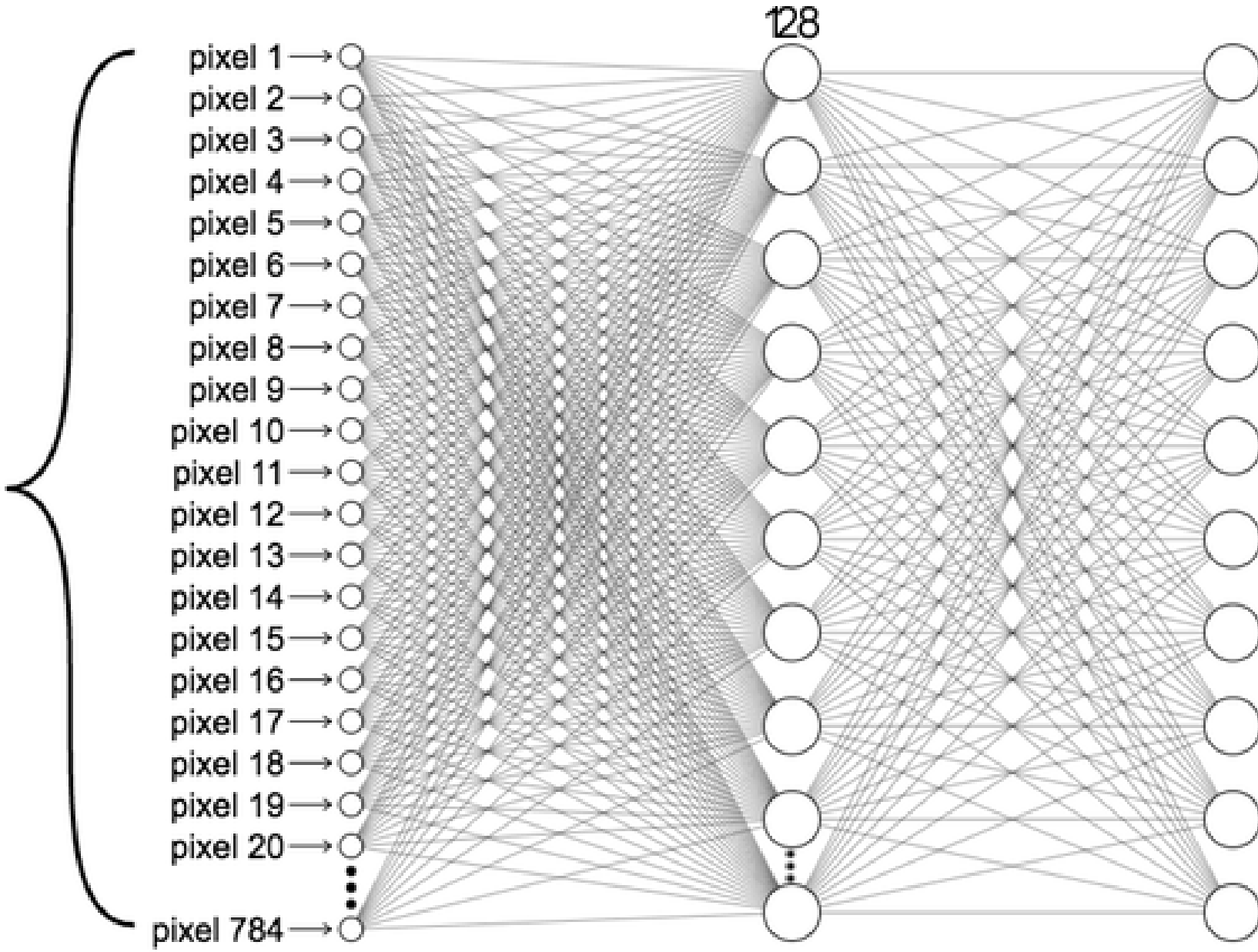
N x M

N*M x 1

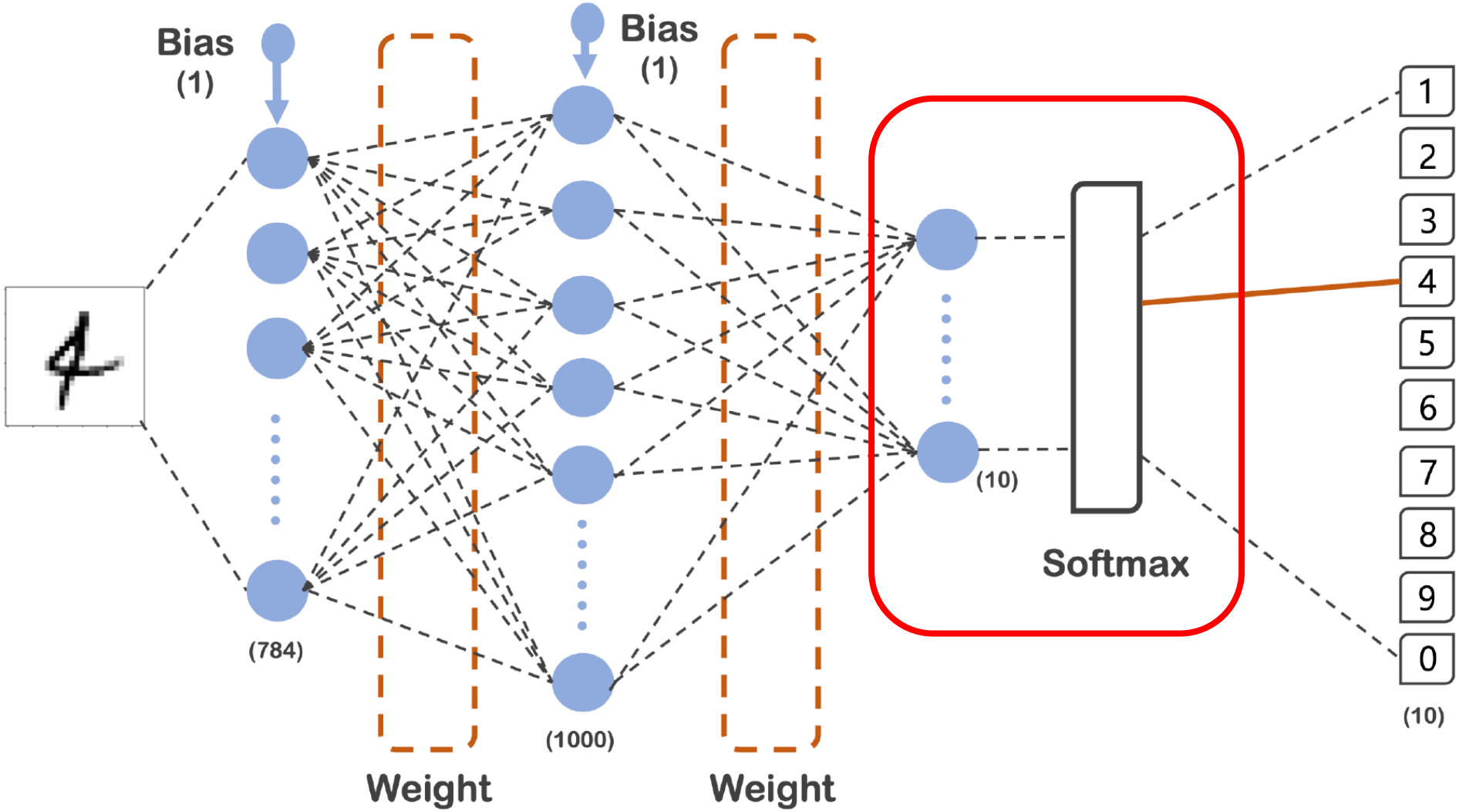


데이터 전처리 과정

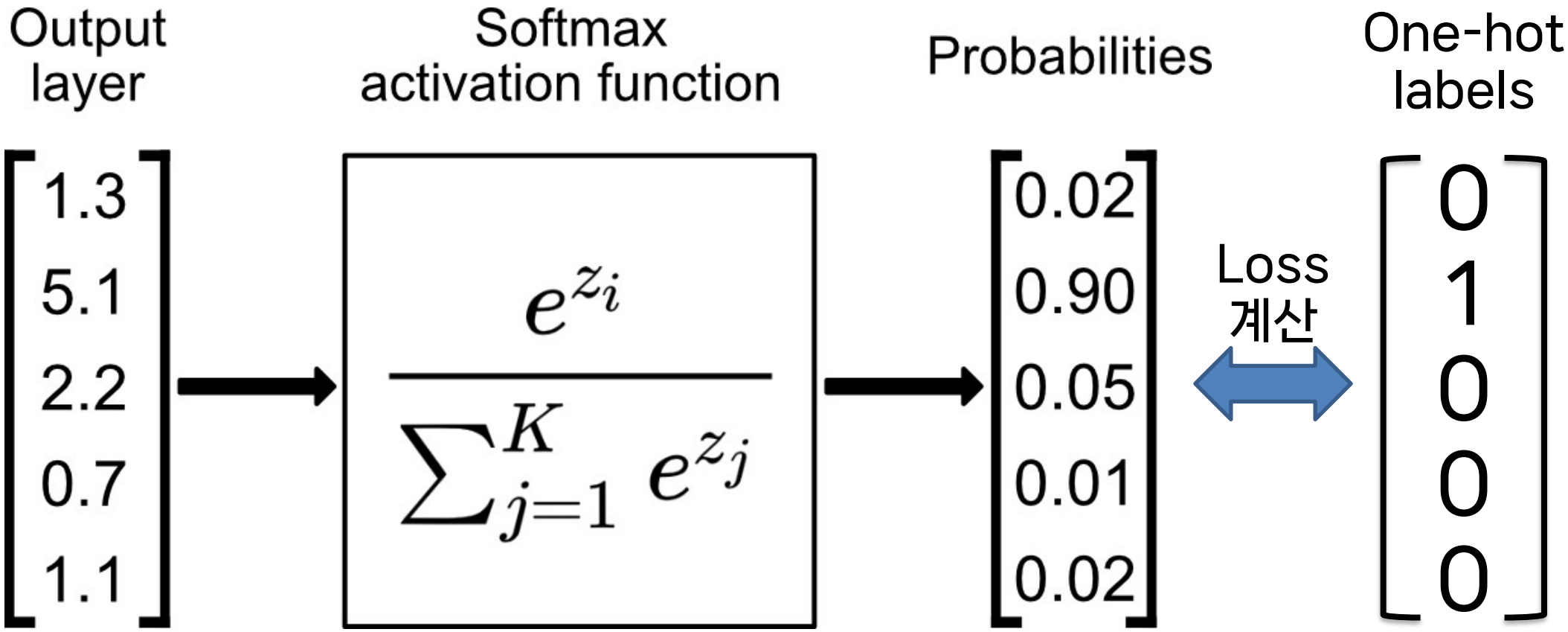



pixel 1
pixel 2
pixel 3
pixel 4
pixel 5
pixel 6
pixel 7
pixel 8
pixel 9
pixel 10
pixel 11
pixel 12
pixel 13
pixel 14
pixel 15
pixel 16
pixel 17
pixel 18
pixel 19
pixel 20
pixel 784
128

MNIST 분류를 위한 deep neural network

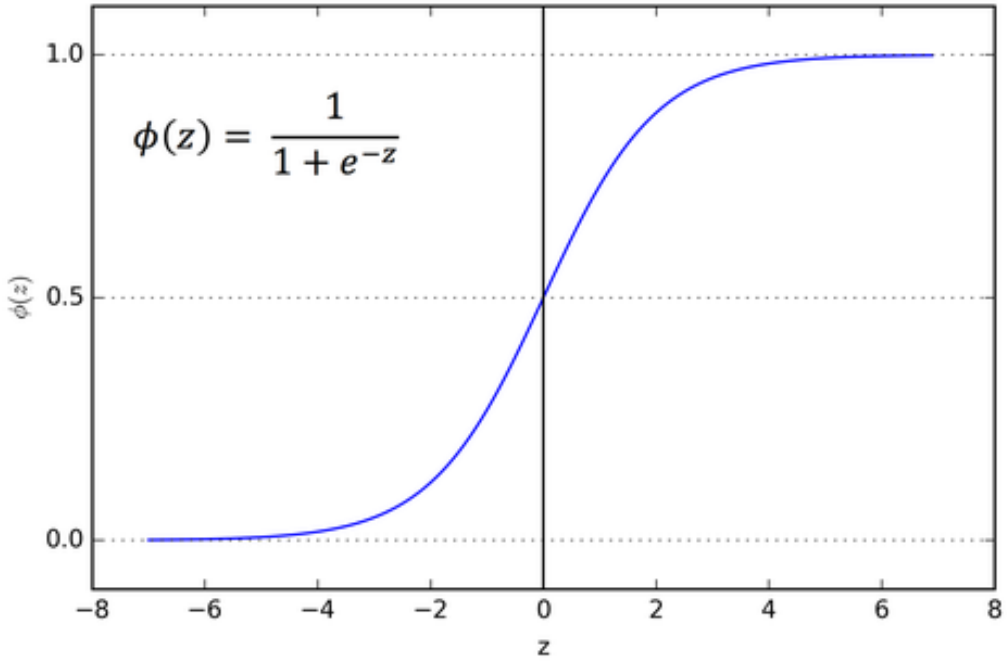


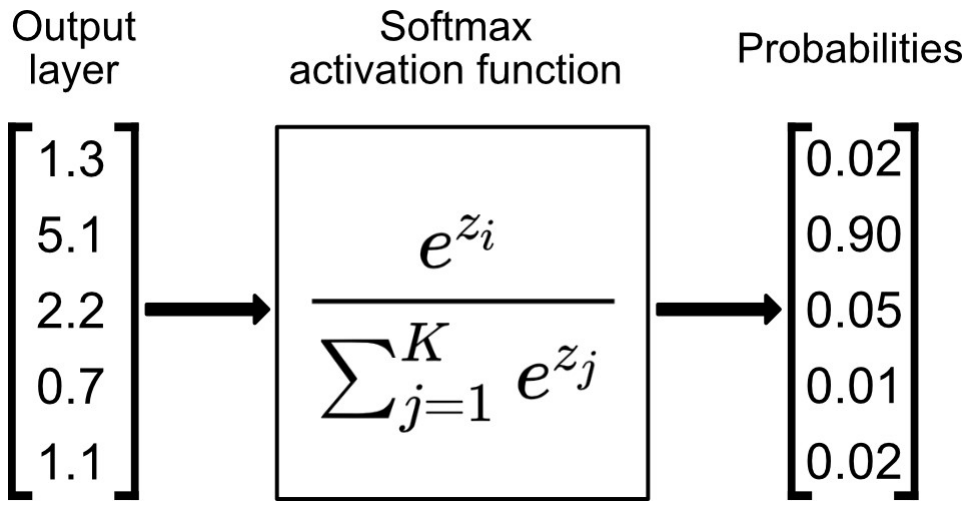
Multi-class 분류를 위한 Softmax



Sigmoid vs Softmax

- Sigmoid는 입력된 실수 값을 0~1 사이로 변환해주는 non-linear function
- Softmax는 입력된 N개의 실수 값을 확률로 변환해주는 non-linear function
- Binary classification-Sigmoid / Multi classification-Softmax





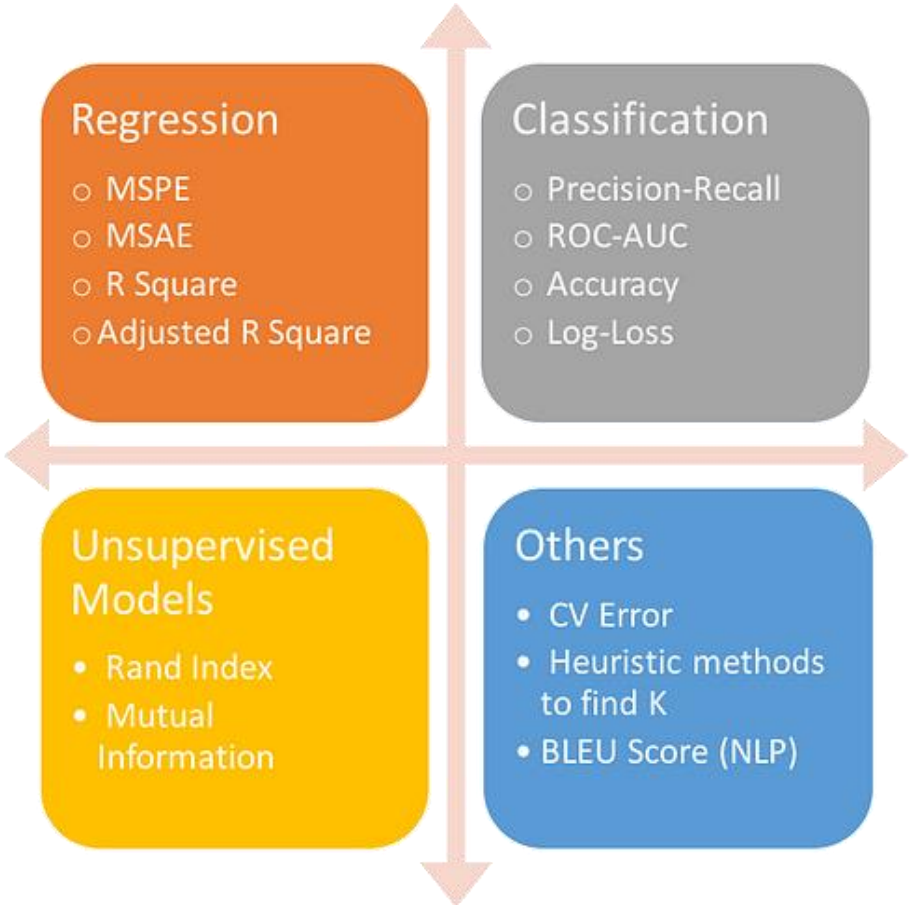
머신러닝 모델의 평가 metric

Metric 이란?

- 모델을 통해 해결하고자 하는 특정 task(분류, 회귀, ...)를 얼마나 잘 수행하는지 평가하기 위한 척도!
- 각각의 task에 적합한 metric들이 존재함
- Loss는 모델의 파라미터를 직접적으로 업데이트하기 위한 기준, metric은 모델이 직접적으로 수행하는 task에 대한 성능을 파악하기 위해 사용
- Metric이 곧 Loss function이거나, 복합적으로 평가하기 위해 여러 metric을 사용하는 경우도 있음
- 분류 문제의 경우

  Loss: Binary-Crossentropy

  Metric: Accuracy, F1 score



MNIST 분류 모델의 평가 metric

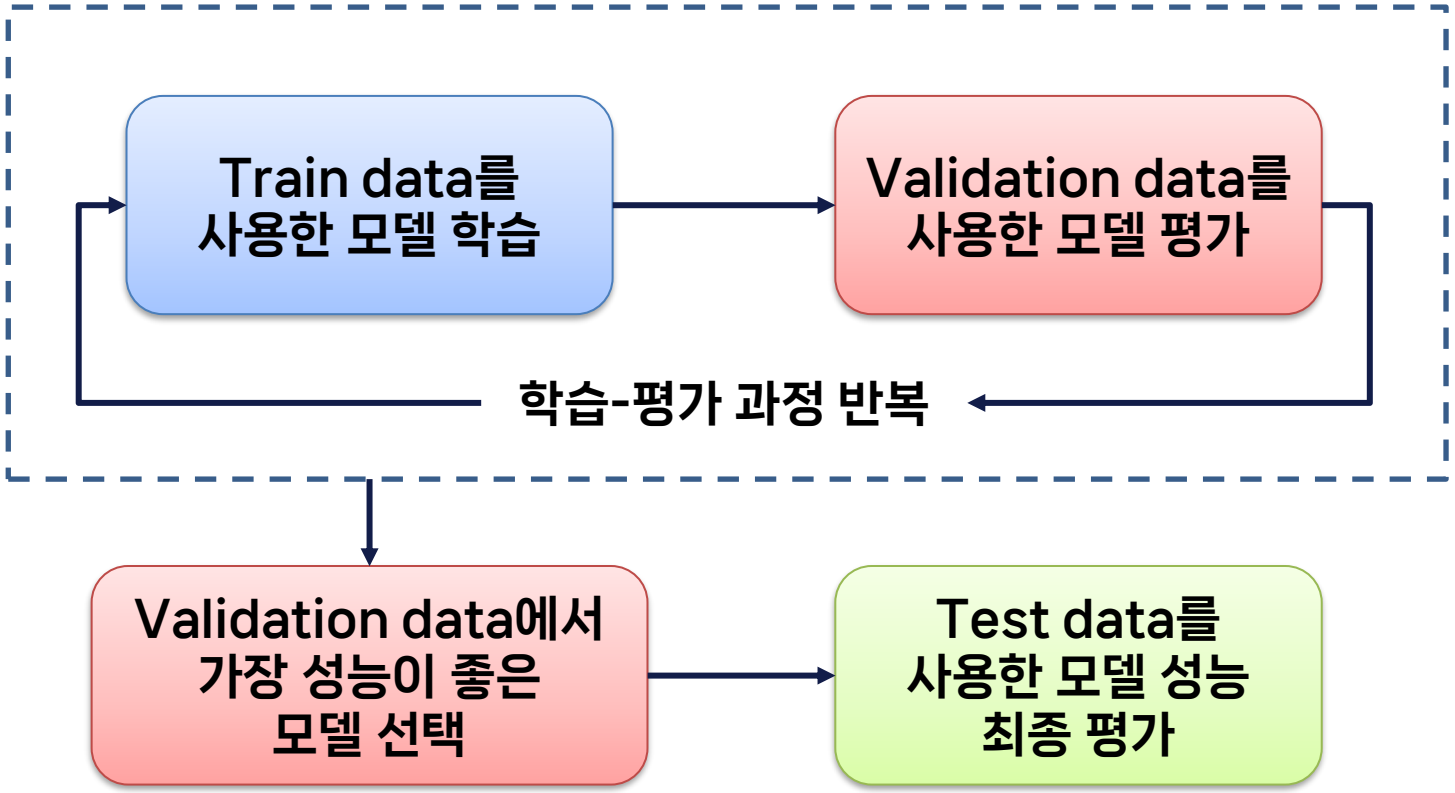
$$\text{Accuracy}(\%) = \frac{\text{Prediction과 Label이 일치하는 데이터 수}}{\text{전체 데이터 수}} \text{ X } 100$$

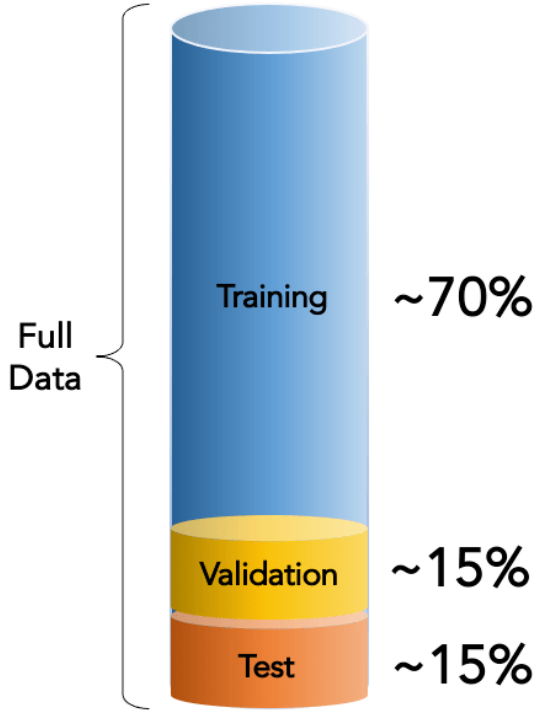
```python
def test(dataloader):
  with torch.no_grad():
    n_correct = 0
    n_samples =0
    for images, labels in dataloader:
        images = images.reshape(-1,28*28)
        outputs = model(images)
        # value, index
        predictions = torch.argmax(outputs,dim=1)
        n_samples += labels.shape[0]
        n_correct += (predictions == labels).sum().item()

    acc = 100 * n_correct / n_samples
    return acc
```

학습 및 평가 데이터 분할

Train? Validation? Test?

- 학습에 사용된 데이터를 모델의 성능을 평가하는데 사용하면 매우 위험!
- 원본 데이터를 분할하여 일부는 학습에 사용하고, 학습에 전혀 관여되지 않은 데이터로 모델의 성능 평가

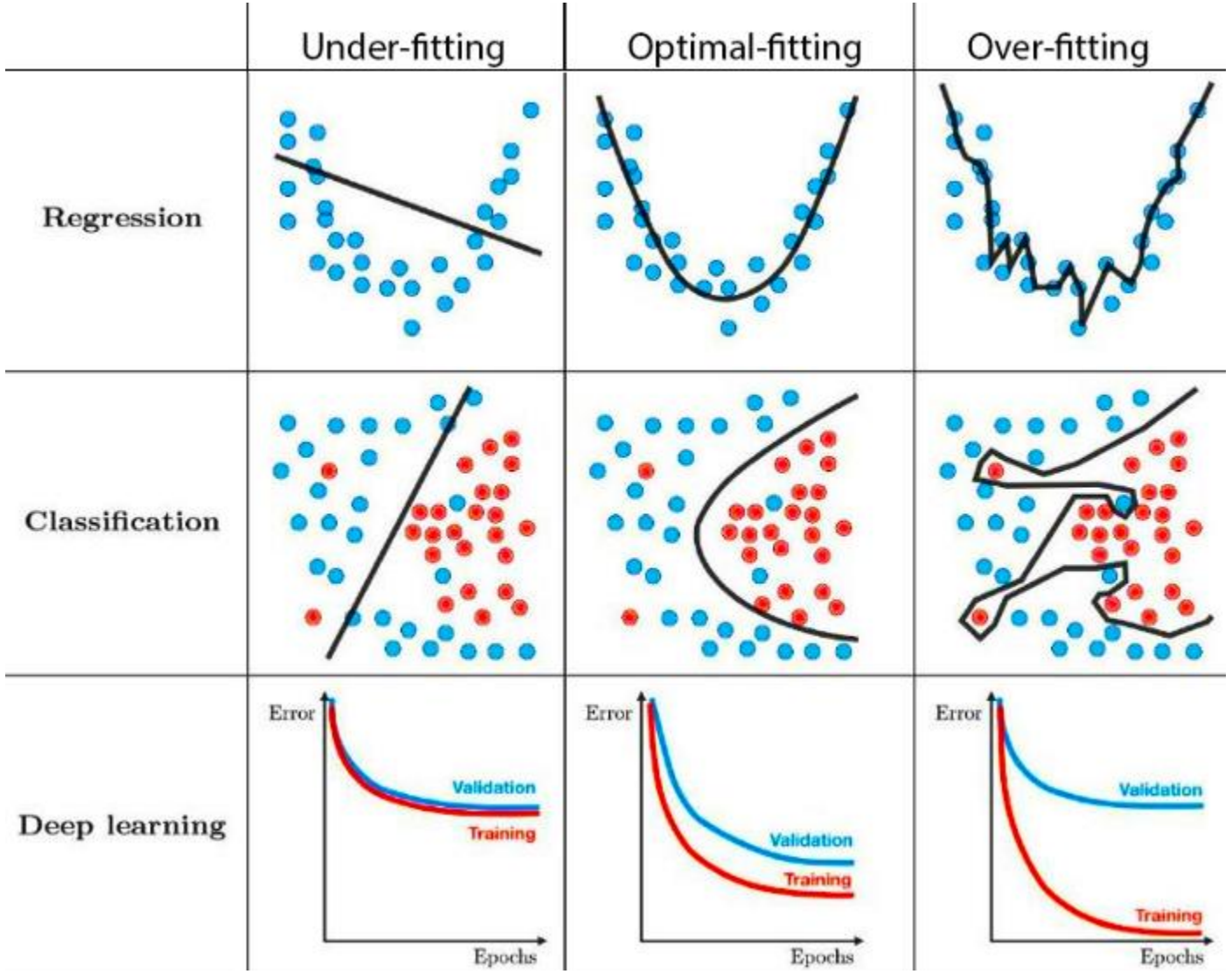




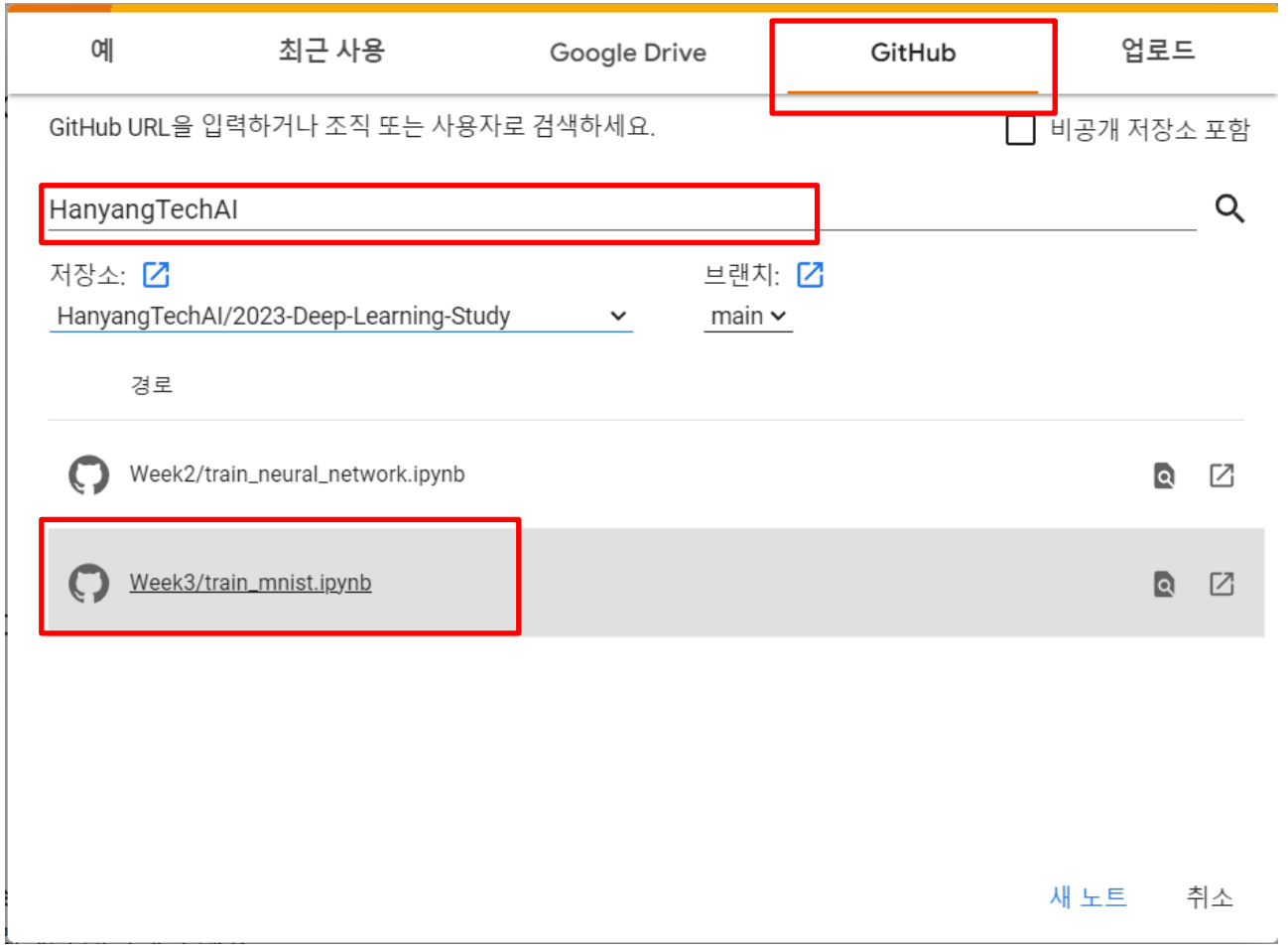
데이터셋을 나누는 이유: 과적합(Overfitting) 방지

Overfitting 이란?

- 모델이 학습에 사용된 데이터 분포와 실제 추론 시에 입력되는 데이터의 분포는 다를 수 있음
- 학습 데이터에만 지나치게 fit된 모델의 경우, 새로운 데이터에 대해서 성능이 떨어질 수 있음
- Validation과 Test 데이터셋이 학습 과정에 영향을 미치지 않아야 공정한 평가가 가능함



Colab에서 github에 업로드된 노트북 불러오기



세 번째 과제: Fashion MNIST 분류 모델 학습하기

Step 1. 예시 노트북을 실행해보며 MNIST 분류 모델 튜닝하기

- 손글씨 데이터를 분류하는 문제를 해결하는 모델을 정의하고, 학습 및 평가 과정을 살펴봅시다
- https://colab.research.google.com/github/HanyangTechAI/2023-Deep-Learning-Study/blob/main/Week3/train_mnist.ipynb#scrollTo=39774407

Step 2. 기존 모델에 새로운 데이터셋(Fashion MNIST) 분류 문제 학습시키기

- 손으로 쓴 숫자 이미지를 분류했던 것 처럼, 다양한 종류의 의류 이미지를 분류하는 모델을 학습시켜 봅시다. 과연 기존 모델이 이 데이터도 잘 분류할 수 있을까요?
- 기존 노트북의 데이터 정의 부분을 변경한 노트북(https://github.com/HanyangTechAI/2023-Deep-Learning-Study/blob/main/Week3/train_fashion_mnist.ipynb)을 바탕으로 조별로 자유롭게 토의해서, 결과를 Github 이슈(https://github.com/HanyangTechAI/2023-Deep-Learning-Study/issues/4)에 업로드해주세요!

With HAI, Fly High

Hanyang Artificial Intelligence