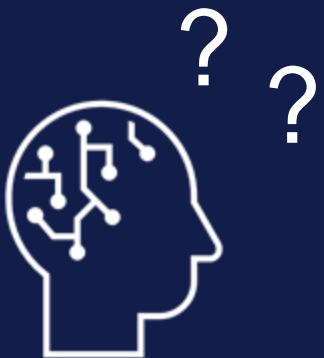


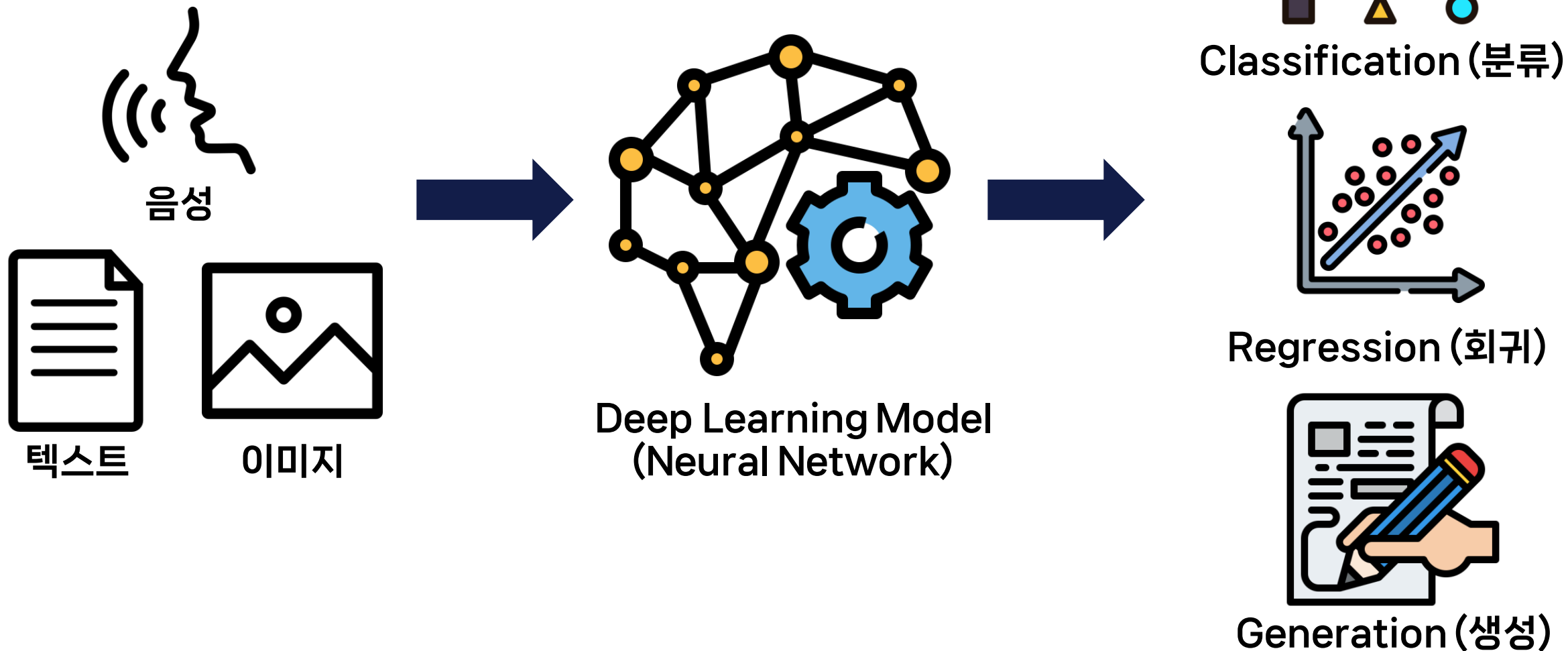
Deep Learning Week 2

How to train deep learning model

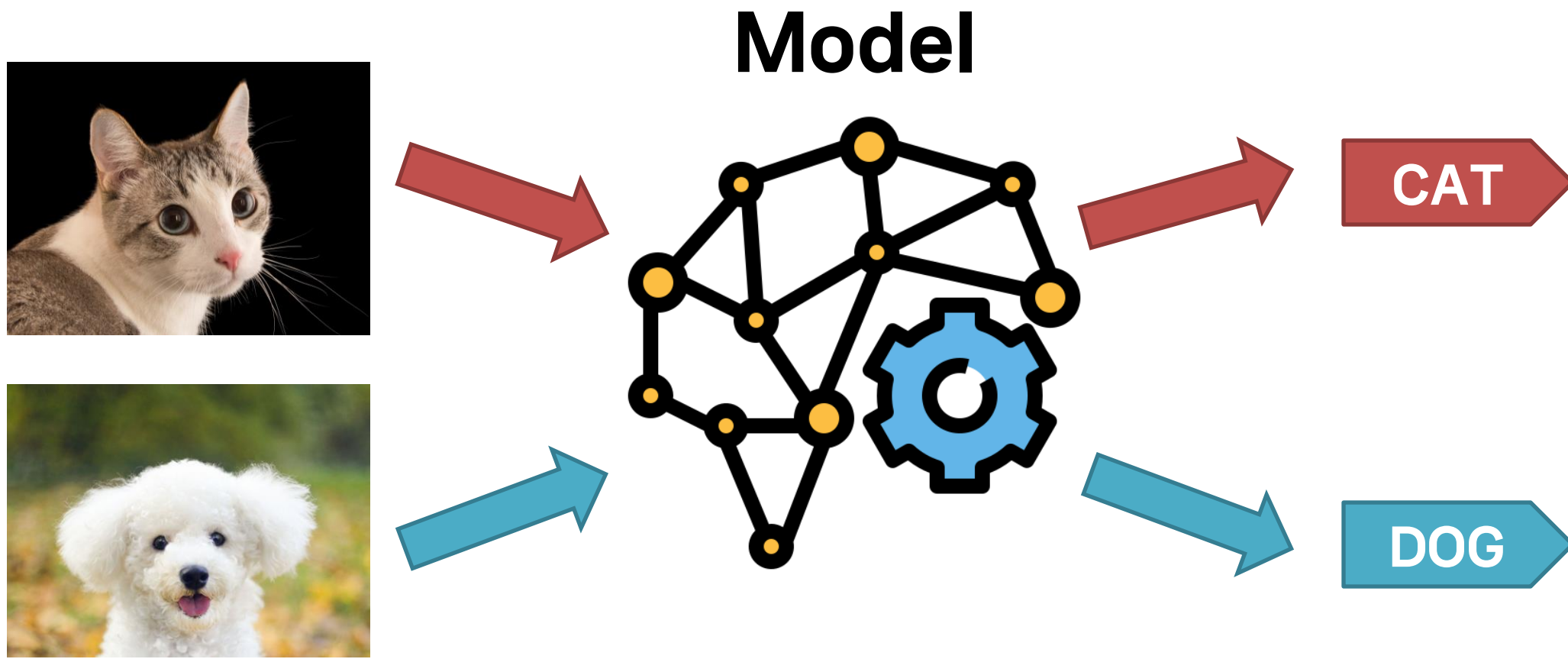
Hanyang Artificial Intelligence Group



Recall: 딥 러닝으로 무엇을 할 수 있을까?



Recall: 고양이 vs 강아지



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, ...

머신러닝 모델 예시: 고양이 vs 강아지

X: 동물 이미지

Y: 분류 라벨

고양이/강아지 분류 모델

$$Y = F(X)$$

$X = \{ \text{images of cat \& dogs} \}$

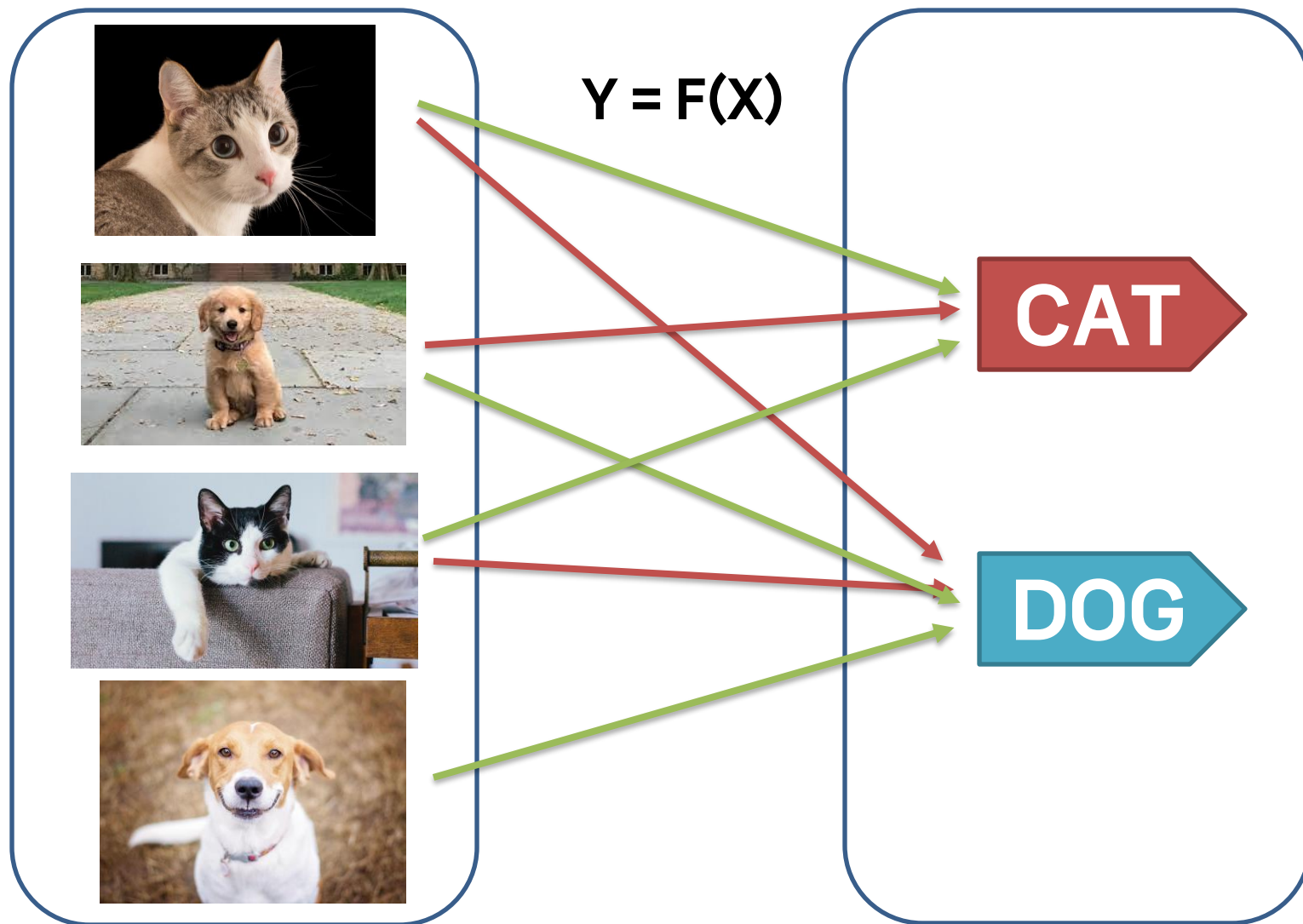
$Y = \{ 0(\text{CAT}), 1(\text{DOG}) \}$

학습 이전

- Accuracy: 25%
- Loss: High

학습 이후

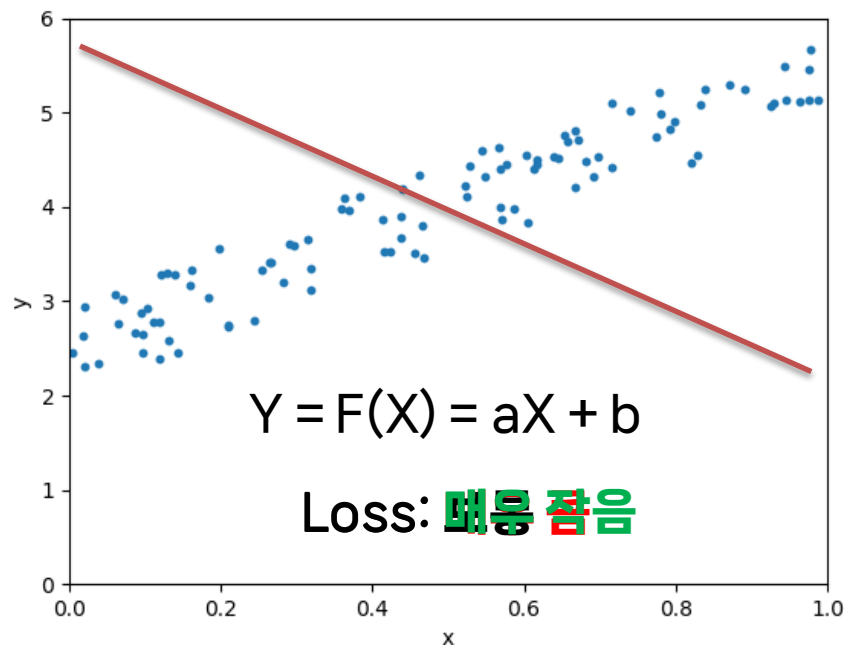
- Accuracy: 100%
- Loss: Low



머신러닝 모델 예시: Linear Regression

Linear Regression이란?

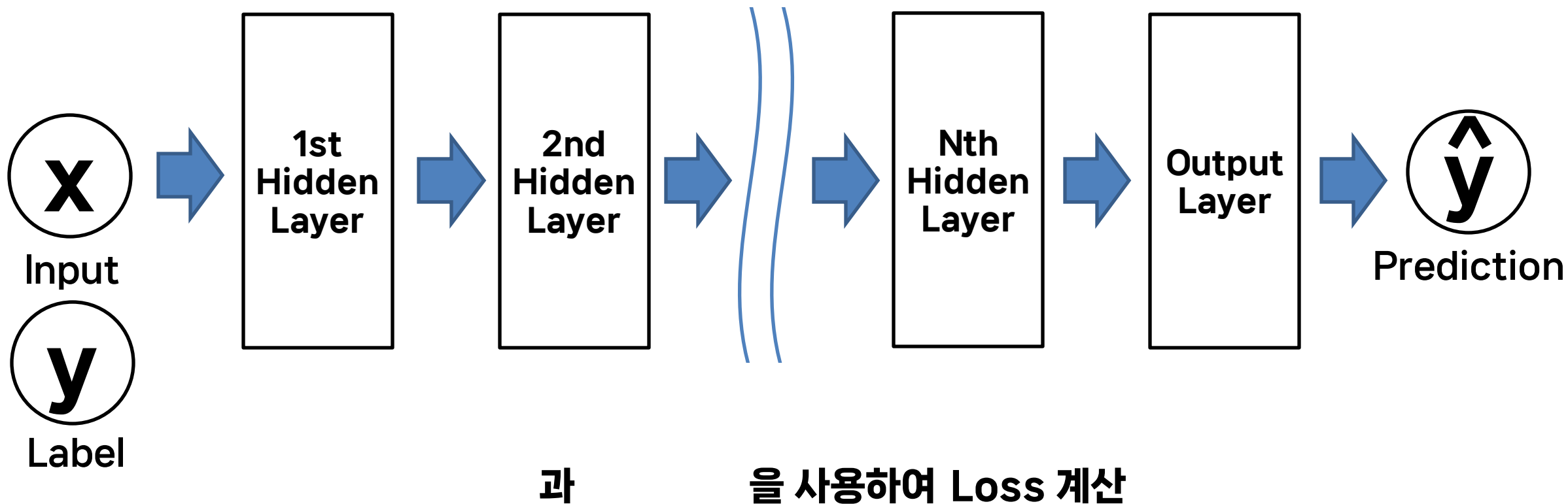
- 주어진 데이터의 분포를 가장 잘 설명할 수 있는 선형 모델(== Linear equation)을 정의하는 것이 목표!
- 학습을 진행하며 모델은 학습 데이터를 더 잘 나타낼 수 있는 형태로 업데이트됨
- Linear Regression의 loss는 각 데이터와 모델 사이의 거리의 평균
- 평균 거리가 작을수록, 모델이 주어진 데이터를 더 잘 표현한다!



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

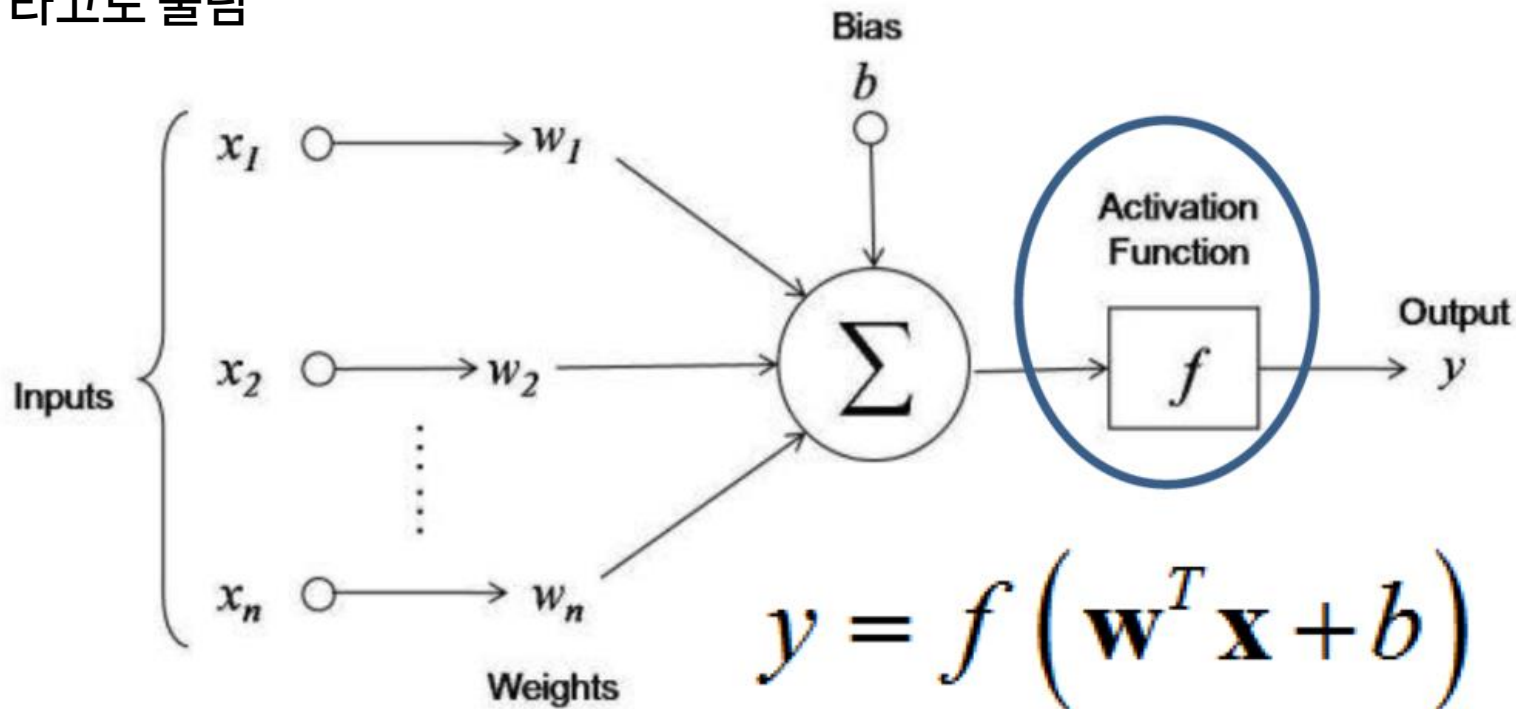
딥 러닝을 구성하는 Neural Network 모델



조금 더 자세히 들여다보면...

딥 러닝 모델을 구성하는 기본 레이어: fully connected layer

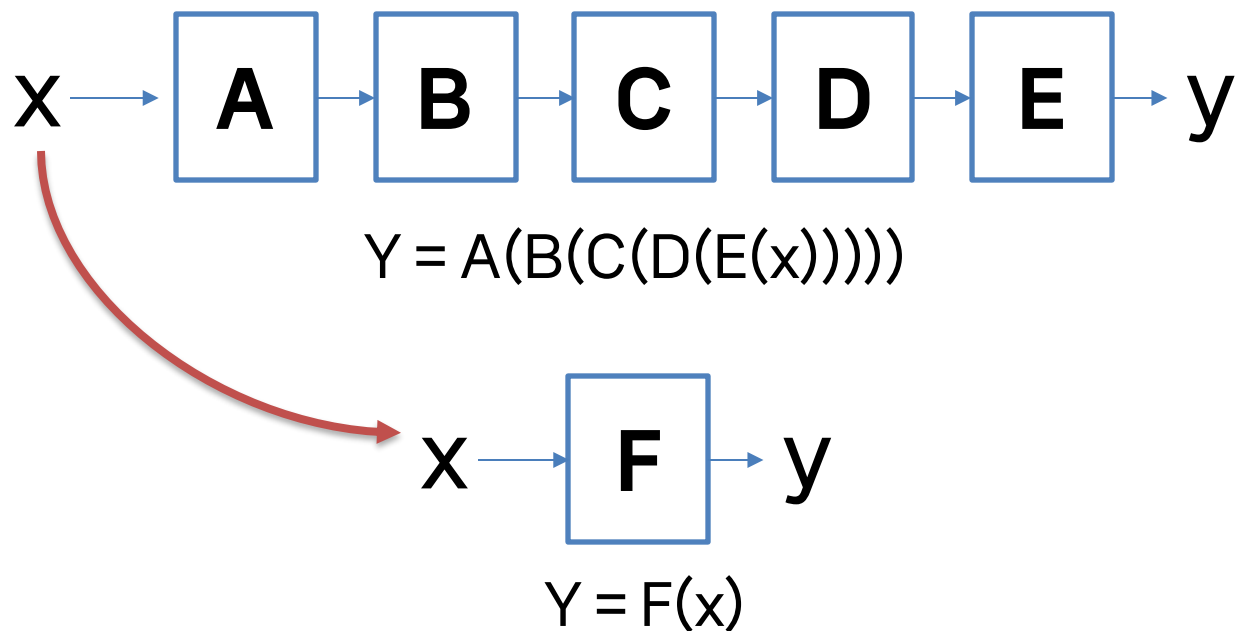
- Input x 와 weight matrix w 를 곱하고, bias vector b 를 더함(Linear equation과 유사)
- 계산된 결과는 activation function을 거쳐 layer의 최종 output 계산
- Dense layer 라고도 불림



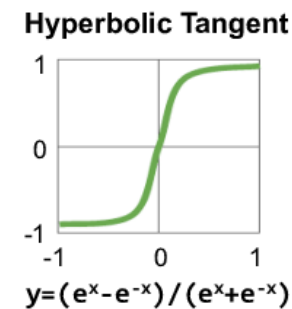
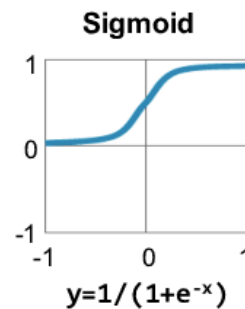
Activation function의 중요성

Non-linear한 activation function

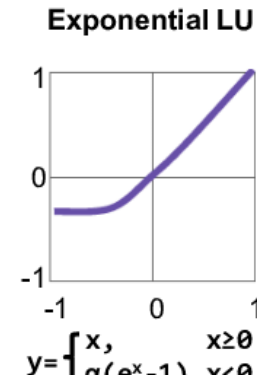
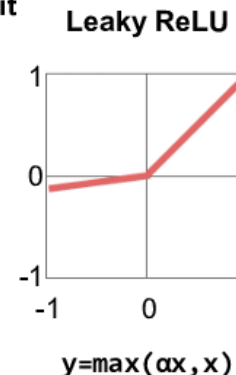
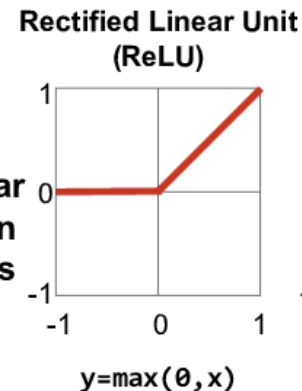
- Input x 와 weight w 를 곱하고, bias b 를 더하는 계산은 linear(선형, 1차원)한 계산
- 만약 activation function으로 non-linear한 함수를 사용하지 않으면, layer를 여러 개 연결해도 결국 하나의 linear equation으로 표현될 수 있음 -> 복잡한 네트워크를 사용하는 의미가 없어짐



Traditional
Non-Linear
Activation
Functions



Modern
Non-Linear
Activation
Functions

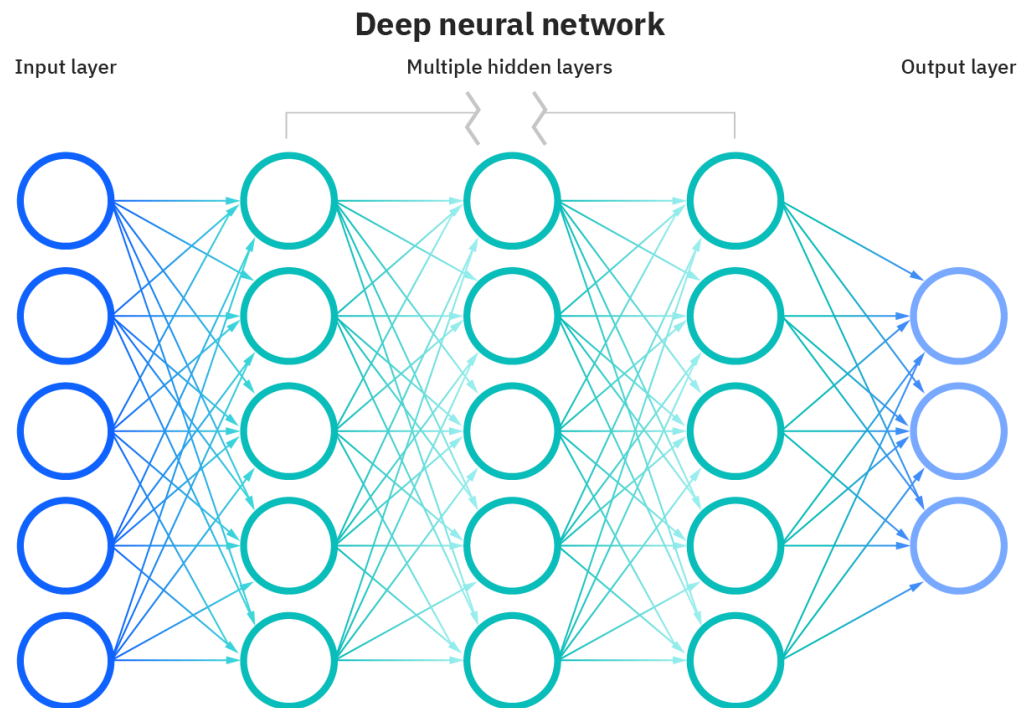
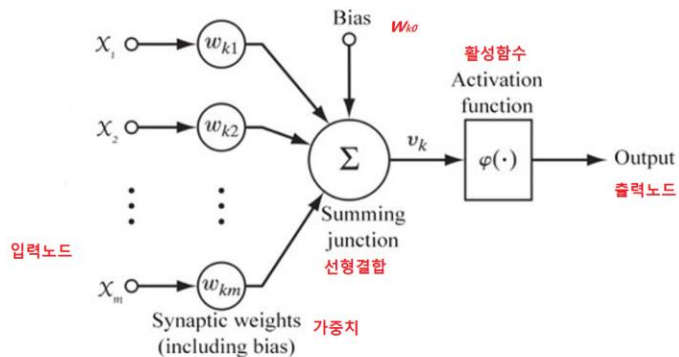
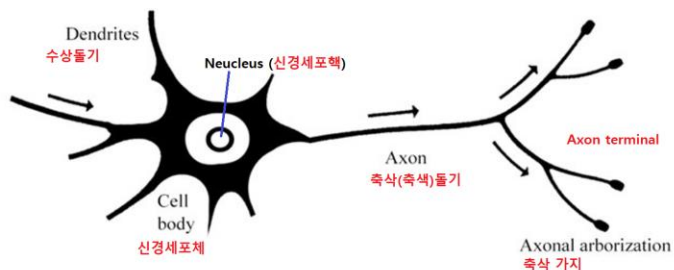


α = small const. (e.g. 0.1)

왜 Neural Network 인가?

인간의 신경망(Neural network)를 모방한 인공 신경망

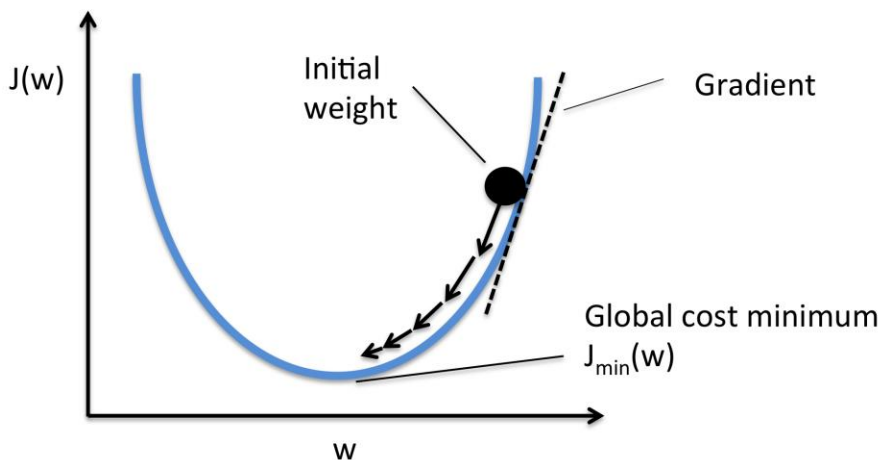
- 수많은 신경 세포(Neuron)들이 서로 연결되어 입출력을 주고받는 인간의 신경망의 구조와 유사하게, 딥 러닝 모델은 수많은 인공 뉴런이 결합하여 다양한 정보를 얻을 수 있도록 설계됨
- 딥 러닝 모델을 인공 신경망(Artificial Neural Network)라고도 부름



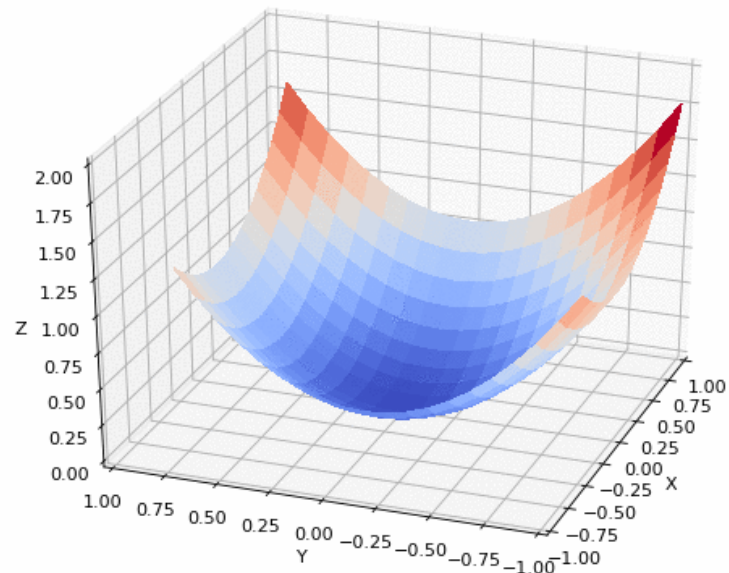
모델 파라미터 업데이트: Optimization

Loss를 최소화하는 최적의 파라미터를 찾자!

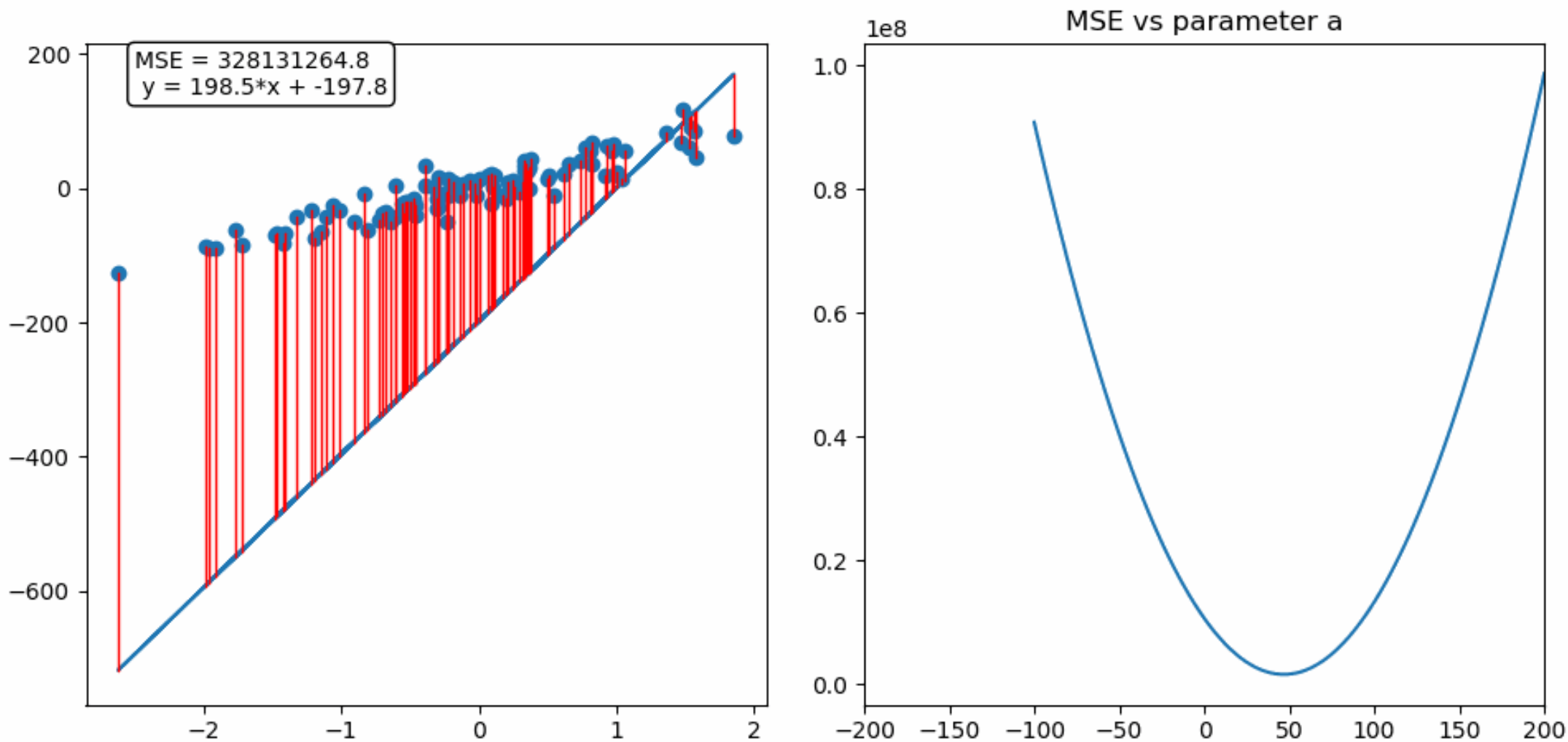
- Optimization: 주어진 데이터와 모델을 활용하여 loss값을 계산한 후, loss값이 작아지도록 하는 최적의 파라미터를 찾는 과정
- Gradient descent: loss를 미분하여 감소하는 방향을 찾고, 해당 방향으로 파라미터를 일정 간격만큼 움직이며 최적의 파라미터를 찾는 optimization algorithm의 한 종류
- Learning rate: gradient descent 과정에서 loss가 작아지는 방향으로 파라미터를 얼마나 많이 변화시킬지 결정하는 파라미터. 너무 작거나 크면 학습이 잘 진행되지 않을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



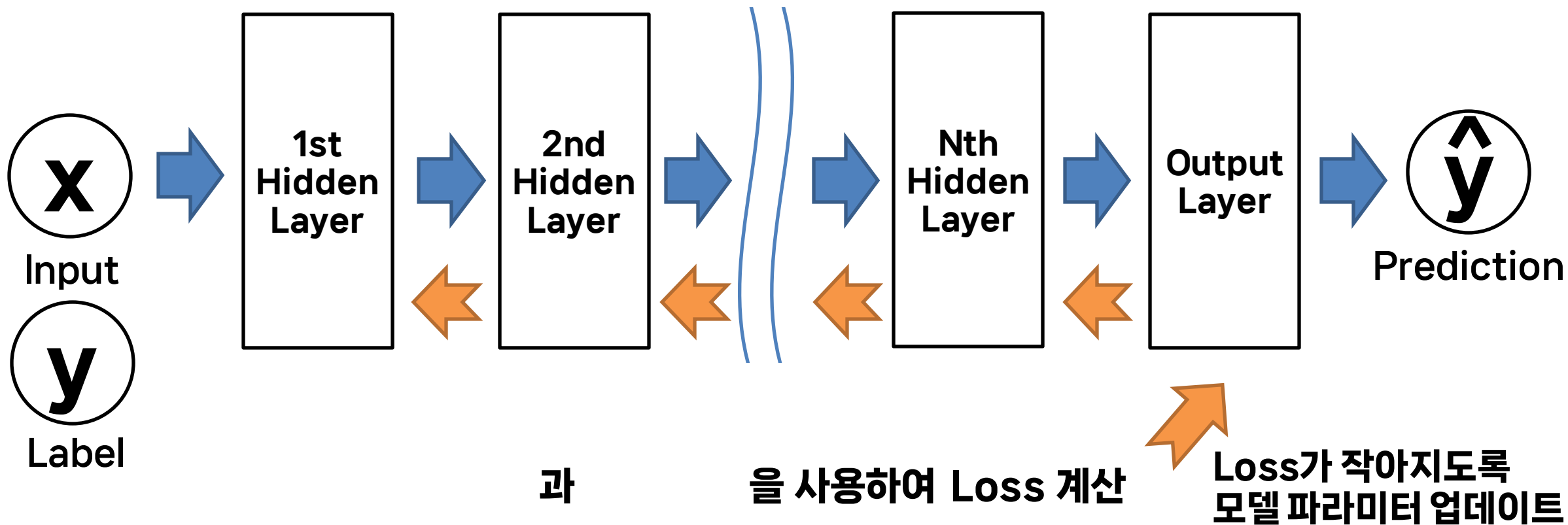
$$\left. \begin{aligned} &\text{repeat until convergence} \{ \\ &\quad \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \\ &\quad \text{(for } j = 1 \text{ and } j = 0) \\ &\} \end{aligned} \right\}$$



Linear regression 모델의 gradient descent 과정



딥 러닝 모델의 back propagation



딥 러닝 학습을 도와주는 Python 라이브러리



- DataFrame 오브젝트를 기반으로 한 데이터 분석
- 정수, 소수, 문자, 시계열 등 다양한 형태의 데이터 분석
- 학습을 위해 데이터를 사전 준비할 때 사용



- 다양한 수학 및 과학 연산 지원
- 벡터, 행렬 계산 및 다양한 수치해석, 선형대수학 관련 함수 지원
- C, Fortran 백엔드로 구현되어 빠른 계산 속도



- 딥러닝 모델, optimization, loss, back propagation 지원
- 모델 계산 과정에서 대규모 행렬 연산을 GPU를 통해 병렬 연산 가능하도록 지원

Google colabatory

Google에서 제공하는 무료 코드 실행 환경

- <https://colab.research.google.com/>
- 무료로 GPU를 사용하여 python 스크립트를 실행하고, 필요한 정보를 기록할 수 있게 해주는 플랫폼

Python Notebook(Jupyter)

- .py 파일을 한번에 실행하는 기존 cli 방식과 다르게, python 스크립트를 원하는 구간 단위로 나누어 실행 가능
- Colab을 통해 python notebook을 생성, 업로드, 실행, 저장, 공유할 수 있음



Colaboratory란?

Colaboratory(또는 줄여서 'Colab')를 사용하면 브라우저에서 Python을 작성하고 실행할 수 있습니다.

- 구성 필요 없음
- GPU 무료 액세스
- 간편한 공유

학생이든, 데이터 과학자든, AI 연구원이든 Colab으로 업무를 더욱 간편하게 처리할 수 있습니다. [Colab 소개 영상](#)에서 자세한 내용을 확인하거나 아래에서 시작해 보세요.

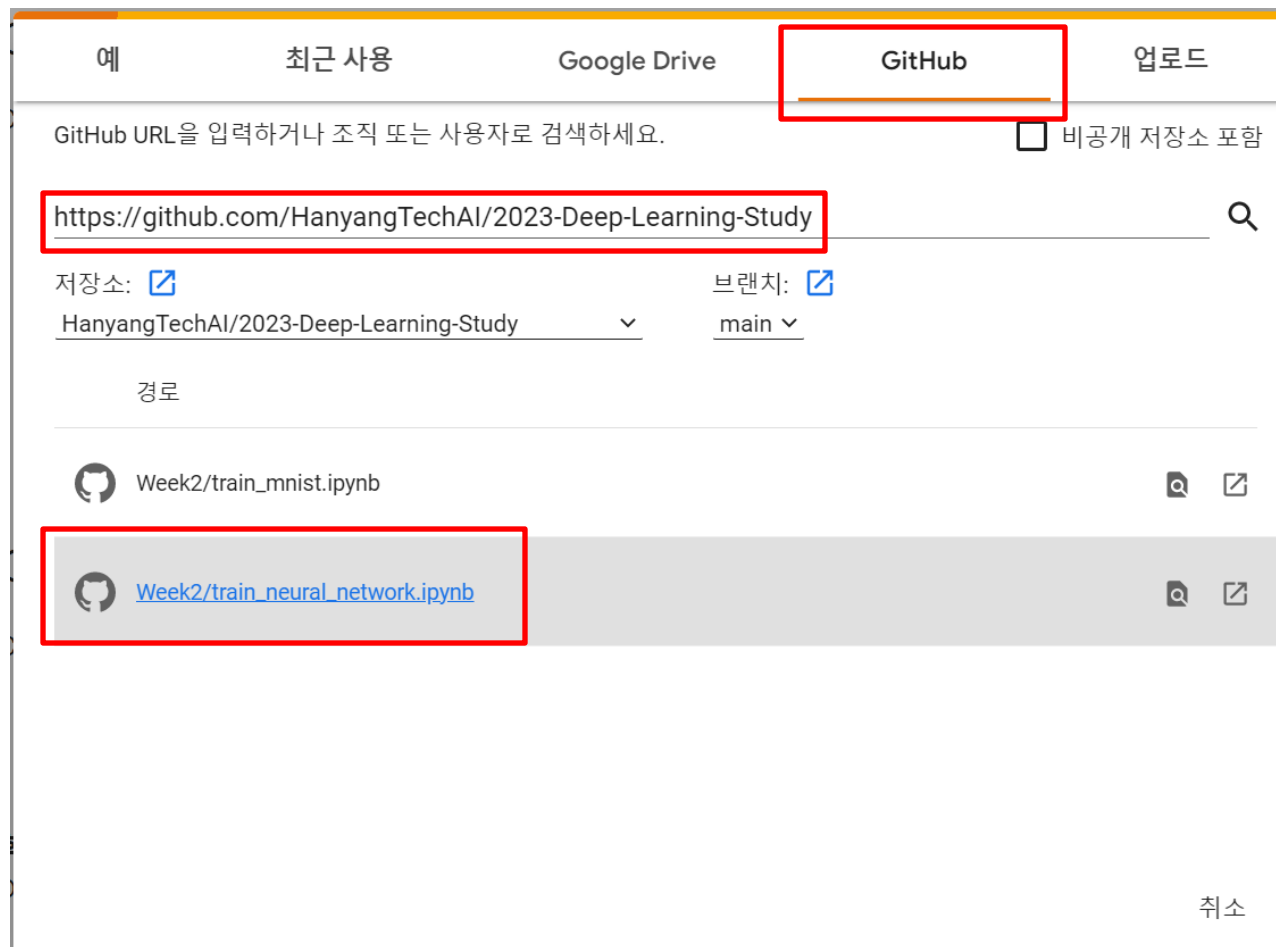
시작하기

지금 읽고 계신 문서는 정적 웹페이지가 아니라 코드를 작성하고 실행할 수 있는 대화형 환경인 Colab 메모장입니다.

예를 들어 다음은 값을 계산하여 변수로 저장하고 결과를 출력하는 간단한 Python 스크립트가 포함된 코드 셀입니다.

```
[ ] seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day
```

Colab에서 github에 업로드된 노트북 불러오기



두 번째 과제: 딥 러닝 모델 학습하기

Step 1. 예시 노트북을 실행해보며 PyTorch 딥러닝 모델 튜닝하기

- 간단한 binary classification 문제를 해결하는 모델을 정의하고, 학습 및 평가 과정을 살펴보기
- https://colab.research.google.com/github/HanyangTechAI/2023-Deep-Learning-Study/blob/main/Week2/train_neural_network.ipynb?hl=ko#scrollTo=50d55775

Step 2. 새로운 데이터셋에 적합한 모델 고르기

- 다른 형태의 데이터셋을 구경해 봅시다. 과연 기존 모델이 이 데이터도 잘 분류할 수 있을까요?
- 기존 노트북의 가장 코드 블록(데이터 생성하는 부분)만 변경해서 실행하며, 모델이 데이터를 잘 분류할 수 있는지 확인해 주세요!
- 조로 자유게토의해서, 결과를 Github 이 (<https://github.com/HanyangTechAI/2023-Deep-Learning-Study/issues/3>)에 업로드해주세요!

With HAI, Fly High

Hanyang Artificial
Intelligence