기계 학습(ML: Machine Learning) 교육 자료

Natural Language Processing

이성희

2022-04-12

목차

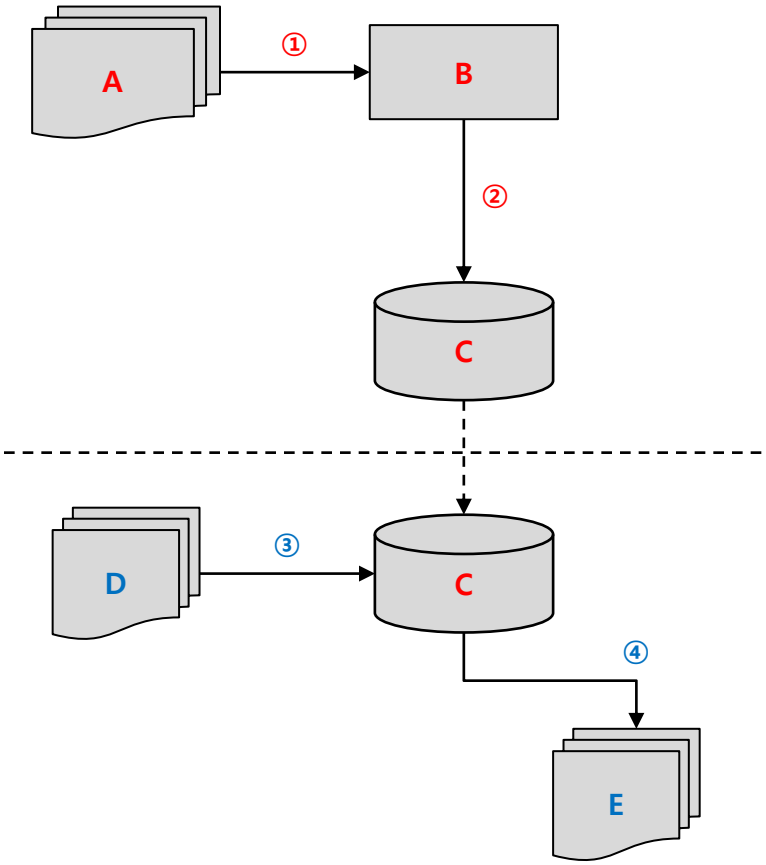
1-1. 기계 학습 개요

- 기계 학습(ML: Machine Learning, 머신 러닝)이란?
  - 기계 학습은 일종의 소프트웨어이다.
  - 규칙 기반 프로그래밍은 개발자가 명확한 규칙을 토대로 프로그래밍하는 것을 뜻한다.
    - 이런 경우에는 이렇게, 저런 경우에는 저렇게, ...
    - 하지만, 이러한 규칙이 무수히 많은 경우에는 어떻게 프로그래밍할 것인가?
  - 사람의 힘으로 처리하기 힘든, 무수히 많은 규칙이 필요한 경우에는 프로그래밍이 매우 어려워진다.
  - 이를 해결하기 위해 제안된 이론이 'Machine Learning'이다. (1959 – Arthur Samuel)
    기계 학습의 정의는 다음과 같다.

    * 개발자가 직접 규칙을 정의하고 프로그래밍하는 것이 아니라, 모델 스스로가 규칙들을 학습

    * 모델이란, 데이터 또는 현상에서 자동으로 규칙들을 학습하는 프로그램

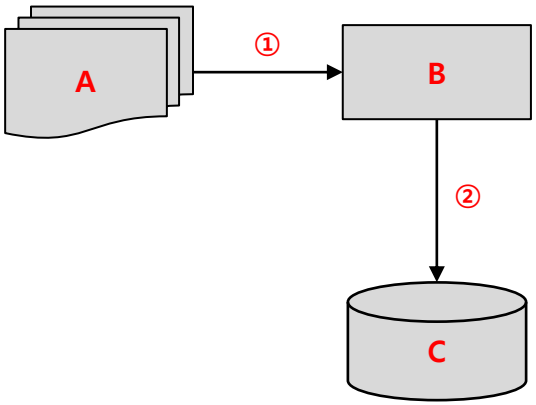
1-2. 기계 학습 프로세스 (1/4)



- ①번과 ②번 과정
  - A를 B에 넣는다.
  - B를 실행시킨다.
  - B가 C를 생성한다.

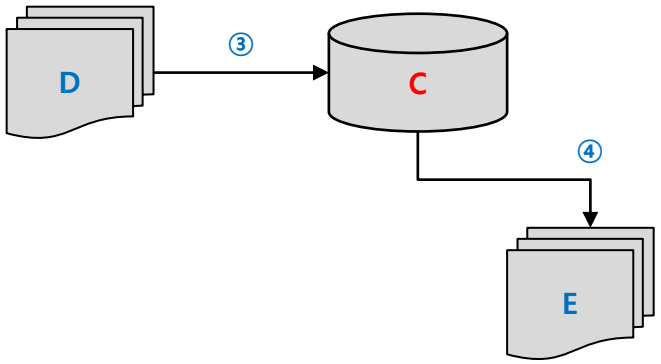
- ③번과 ④번 과정
  - D를 C에 넣는다.
  - C가 E를 예측한다.

1-2. 기계 학습 프로세스 (2/4)



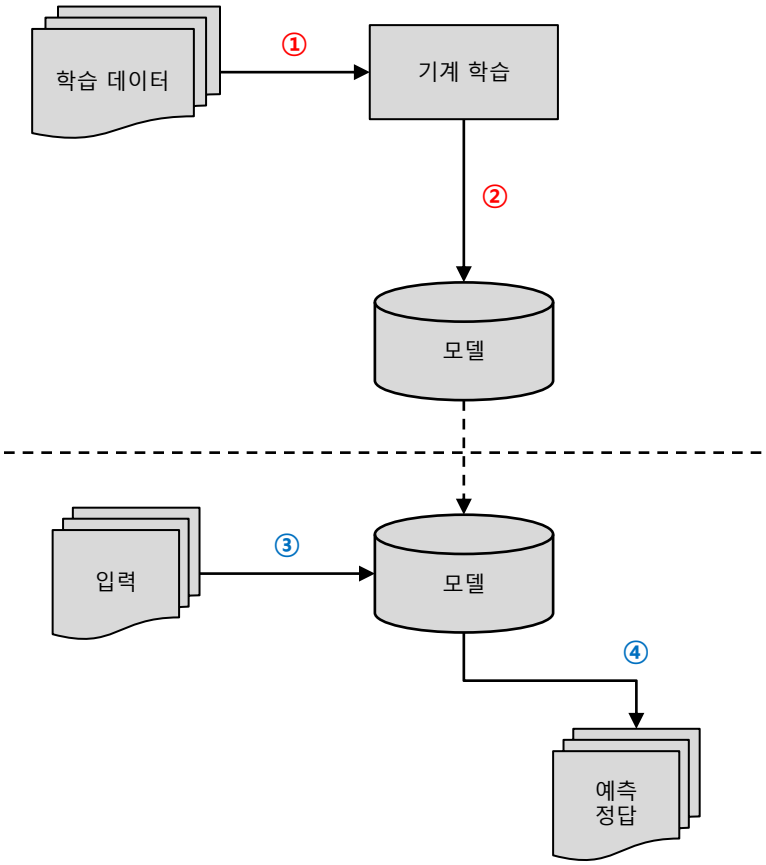
- A : 학습 데이터 (training data)
  - ML에서 학습을 위한 기본 데이터, 자동으로 규칙을 찾아내기 위해서는, 대용량의 데이터가 필요
- B : 기계 학습 (machine learning)
  - 실제 기계 학습을 수행하는 부분
  - 개발자가 해결하려는 문제와 학습 데이터(A)를 파악하여, 알고리즘과 구성을 설계하고 기계 학습을 위한 실질적인 프로그램을 구현하는 부분
- C : 모델 (model)
  - 기계 학습의 결과물, B에서 기계 학습이 완료되면, 모델(C)이 생성되고 해결하려는 문제는 이 모델의 성능에 영향을 받는다.
  - 사용자 입장에서 실제로 문제를 푸는 프로그램은 모델이고, 이 모델을 생성하기 위해 개발자가 프로그래밍하는 부분이 B이다.

1-2. 기계 학습 프로세스 (3/4)



- D : 입력 (input)
  - 사용자가 정답을 알고 싶은 입력
- E : 예측 정답 또는 결과 (prediction 또는 output)
  - 모델(C)이 입력(D)에 대하여 예측한 정답(E)

1-2. 기계 학습 프로세스 (4/4)



- ①번과 ②번 과정은 개발자가 기계 학습을 수행하여 프로그램을 개발하는 과정

- ③번과 ④번 과정은 개발 완료된 프로그램을 이용하여 사용자가 문제를 해결하는 과정

목차

2-1. 모델 학습 방법 (1/4)

- 지도 학습 (Supervised Learning)
  - 학습 데이터에 이미 정답(Label)이 부착된 데이터
  - 지도 학습에 사용되는 학습 데이터는 학습의 입력으로 데이터(Input data)와 그 입력 데이터의 정답(Label)로 구성
  - 영화 리뷰 데이터의 지도 학습 긍/부정 분류, 학습 데이터 예시

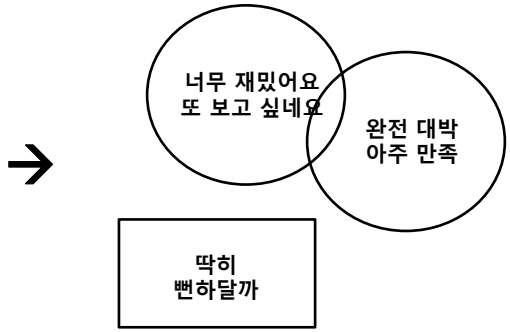
입력 (Input data)	정답 (Label)
너무 재밌어요. 또 보고 싶네요.	긍정
완전 대박 스토리도 좋고 배우 연기도 좋고 아주 만족만족	긍정
딱히 재밌진 않았던 것 같아요. 으음... 스토리가 좀 너무 뻔하달까?	부정
...	...

  - 지도 학습으로 긍/부정 분류 문제를 해결할 경우, 학습된 모델은 새로운 리뷰 데이터에 대하여 긍정 또는 부정에 대한 정답을 예측하여 사용자에게 반환

2-1. 모델 학습 방법 (2/4)

- 비지도 학습 (Unsupervised Learning)
  - 지도 학습과는 반대로 정답(Label)이 부착되지 않은 학습 데이터를 사용
  - 비지도 학습에 사용되는 학습 데이터는 학습의 입력인 데이터(Input data)로만 구성
  - 영화 리뷰 데이터의 비지도 학습, 학습 데이터 예시

입력 (Input data)
너무 재밌어요. 또 보고 싶네요.
완전 대박 스토리도 좋고 배우 연기도 좋고 아주 만족 만족
딱히 재밌진 않았던 것 같아요. 으음... 스토리가 좀 너무 뻔하달까?
...

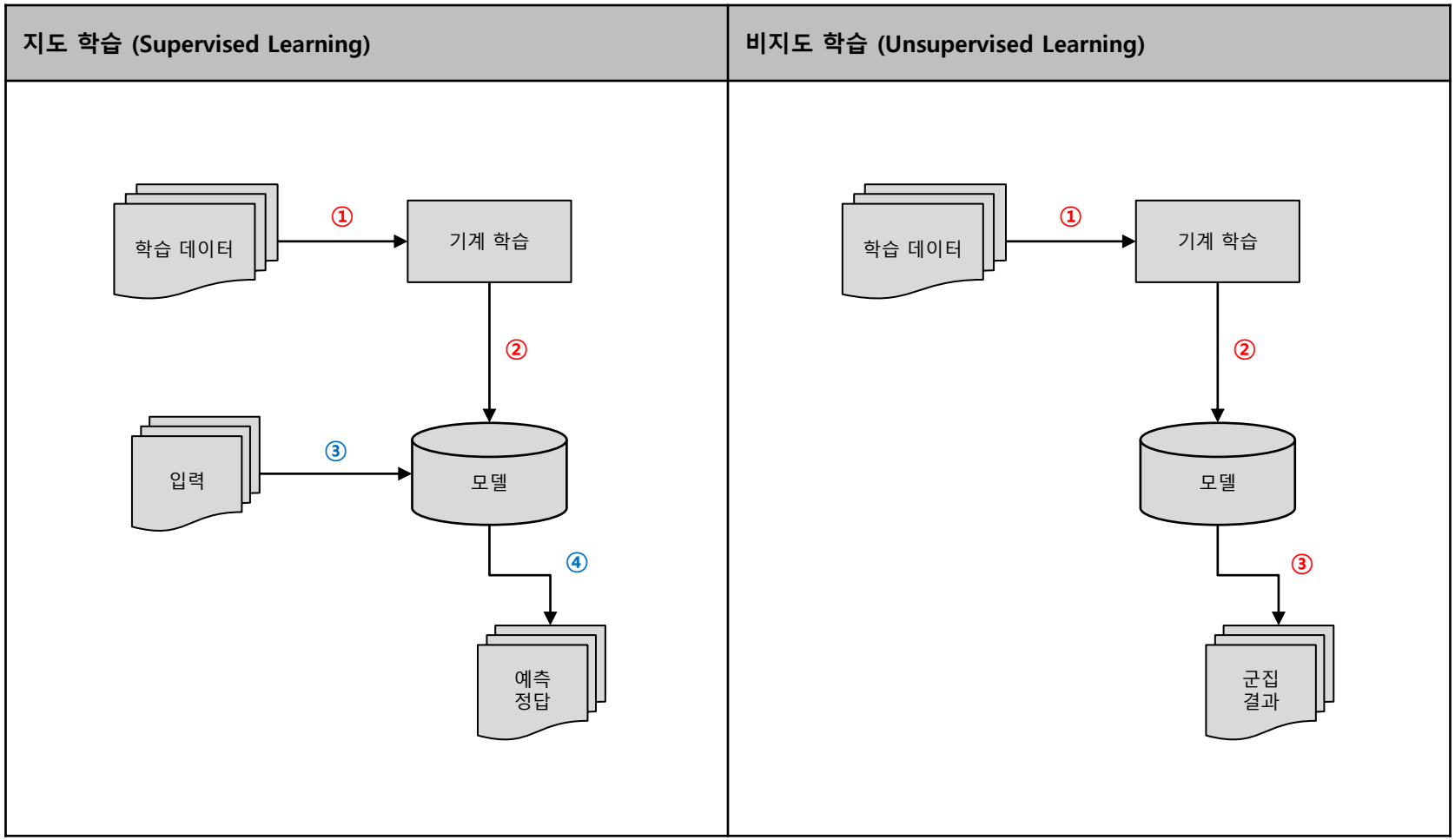


  - 비지도 학습으로 해결 가능한 문제는 대표적으로 군집 또는 클러스터링
    - 비슷한 의미를 내포하고 있는 단어 또는 문장들을 그룹화하여 그룹 별로 나누는 것
    - 정답이 없으므로, 입력 데이터로만 학습

2-1. 모델 학습 방법 (3/4)

	지도 학습 (Supervised Learning)	비지도 학습 (Unsupervised Learning)
학습 방법 목표	• 정답을 알려주면서 학습을 시키는 것 • 입력 데이터와 정답 간에 매핑할 함수를 학습	• 정답을 알려주지 않고 학습을 시키는 것 • 주어진 데이터가 가지고 있는 숨겨진 패턴을 학습
활용	• 회귀 (Regression) • 분류 (Classification)	• 클러스터링 (Clustering) • 특성 학습 (Feature learning)
학습 과정	고양이 사진(Input data)을 주면서 이 사진은 고양이(Label)입니다. 병아리 사진(Input data)을 주면서 이 사진은 병아리(Label)입니다. ... (지도 학습 완료)	고양이, 병아리, 기린, 호랑이, 공룡 사진을 보여주면서 어떤 동물인지는 알려주지 않고 비지도 학습 진행 고양이, 고양이, 병아리, 기린, 공룡, 호랑이, 호랑이, 기린, 병아리, ... (비지도 학습 완료)
학습 결과	고양이 사진을 주면서, 이 사진은 어떤 동물이야? → 고양이 호랑이 사진을 주면서, 이 사진은 어떤 동물이야? → 고양이 닭 사진을 주면서, 이 사진은 어떤 동물이야? → 병아리	다리가 4개이고 머리, 몸통, 꼬리가 일직선 → 고양이와 호랑이 그룹 다리가 4개지만, 목이 위로 긴 → 기린, 공룡 그룹 다리가 두개이고 몸통이 둥그런 → 병아리 그룹

2-1. 모델 학습 방법 (4/4)


지도 학습 (Supervised Learning)
학습 데이터
①
기계 학습
②
입력
③
모델
④
예측
정답
비지도 학습 (Unsupervised Learning)
학습 데이터
①
기계 학습
②
모델
③
군집
결과

2-2-1. 지도 학습 활용 – 회귀 (Regression)

- 통계학에서의 회귀 분석
  - 관찰된 연속형 변수들에 대해 두 변수 사이의 모형을 구한 뒤, 적합도를 측정해 내는 분석 방법

- 기계 학습에서의 회귀 분석
  - 임의의 값을 예측하는 문제
    - 공부한 시간을 기준으로 시험 성적을 예측
  - 회귀를 쉽게 설명하기 위해 선형 회귀로 예를 들면, 선형 회귀에서 '선형'은 1차원 직선을 뜻하고
    이 직선이 두 변수 사이의 모형에 해당하며, 해당 직선이 올바른지 검증하는 것이 적합도를 측정하는 것과 동일
  - 두 개 이상의 변수들 간의 관계식을 찾아내고, 이 관계식의 적합도를 검증하는 통계 기법
    - 관계식 : 두 개 이상의 변수들을 가장 잘 표현할 수 있는 함수 (수학적 의미의 함수)
  - 선형 회귀 분석의 결과값은 실수 값이며, 연속성을 갖는다.
    (직선의 방정식의 그래프를 생각하면 이해하기 쉽다.)

2-2-2. 지도 학습 활용 – 선형 회귀 (Linear Regression) (1/2)

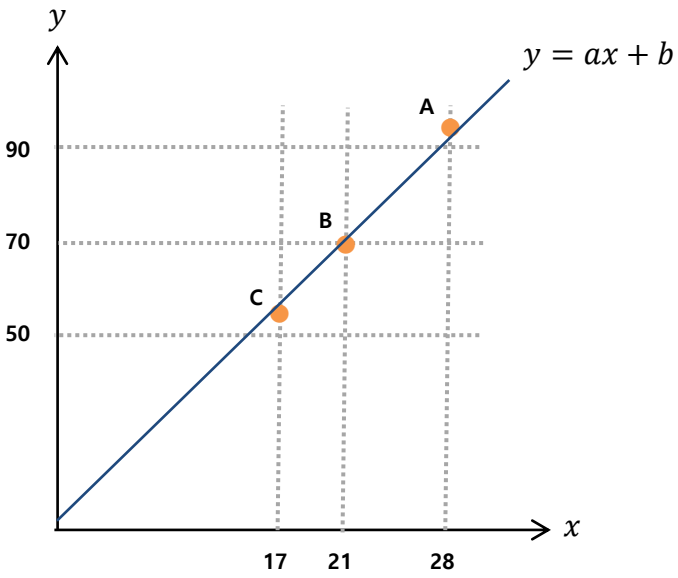
- 선형 회귀 (Linear Regression)
  - 회귀 (Regression) : 변수들 간의 관계식을 찾아내고, 이 관계식의 적합도를 검증하는 통계 기법
    - 선형 회귀의 '선형'은 직선을 뜻하므로, 두 변수 x, y 간의 관계식은 직선의 방정식 형태이다.
  - x, y 간의 관계에 적합한 선, 이 선을 회귀선이라고 하며, 회귀선이 직선인 경우에는 회귀 직선이라고 부른다.
  - 최종적으로 이 회귀 직선을 구하는 문제를 일반적으로 선형 회귀(Linear regression)라고 부른다.
  - 예를 들면, 공부 시간(x)와 시험 성적(y)의 관계를 나타내는 회귀선은 직선
    - 일반적으로 시험 공부를 많이 하면 성적이 높고, 적게 하면 낮기 때문
  - 회귀 직선은 정답(Label 또는 y)의 범위가 매우 넓다.
    - '직선'이라는 개념은 무수히 많은 점이 이어진 것이기 때문
  - x, y 간의 관계를 가장 잘 나타내는 회귀 직선을 찾고, 이 직선을 이용하여 임의의 x 값에 대하여, 대응하는 y 값을 예측하는 것을 (linear) Regression 문제를 푼다고 한다.

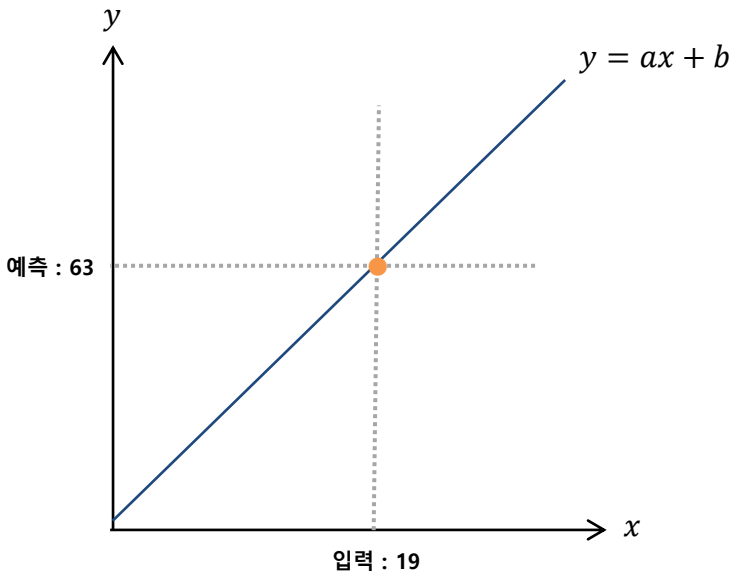
2-2-2. 지도 학습 활용 – 선형 회귀 (Linear Regression) (2/2)

	x (hours)	x (sum)	y (score)
A	11, 8, 9	28	92
B	7, 9, 5	21	70
C	3, 5, 9	17	56
…	…	…	…

- Linear regression을 학습한다는 것은
  학습 데이터의 모든 데이터(좌표 평면 위의 모든 점)을
  가장 잘 표현할 수 있는 하나의 직선을 찾아내는 과정

- 좌표 평면에서 직선은 무수히 많고, 그 무수히 많은 직선 중에서
  모든 데이터를 가장 잘 표현할 수 있는 하나의 직선을 찾는 과정





목차

3-1. 기계 학습에 필요한 3가지 구성 요소

- 3가지 구성 요소
  - 가설 (Hypothesis)
    - 문제를 해결하기 위한 방정식 또는 관계식
      - 최종적으로 모델이 학습 또는 구하려는 대상 (관계식)
      - 다시 말해서 학습이란, 모든 데이터를 가장 잘 표현할 수 있는 하나의 관계식을 찾는 과정
  - 함수 (Function)
    - 손실 함수 (Loss function)

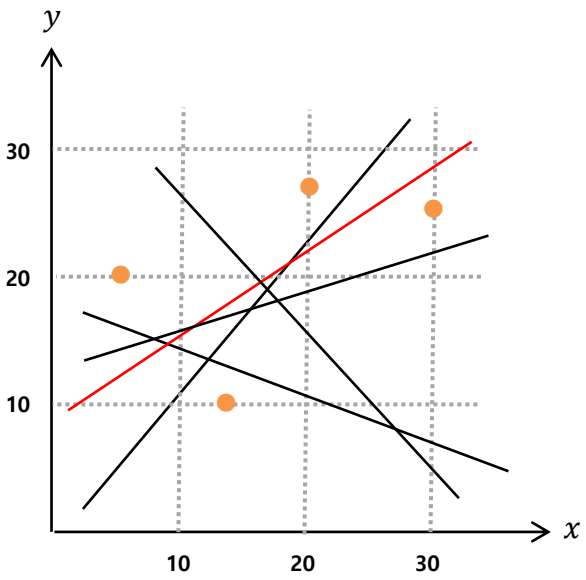
      - 하나의 입력(x)에 대하여, 가설이 예측한 값($y\hat{}$)과 실제 정답(Label 또는 y) 사이의 오차를 계산하는 함수
    - 비용 함수 (Cost function)

      - 전체 학습 데이터에 대하여, 가설이 예측한 값과 실제 정답 사이의 오차를 계산하는 함수

      - 즉, 모든 데이터에 대해 계산한 Loss function의 평균 값(평균 오차)을 뜻한다.
    - 목적 함수 (Objective function)
      - 목적(해결하려는 문제)에 맞게 최댓값 또는 최솟값을 구하는 함수
  - 최적화 (Optimize)
    - 목적 함수를 이용하여, 최댓값 또는 최솟값을 구하는 과정

3-2-1. 기계 학습 과정 예제 – Linear Regression (1/17)

x (input)	y (label)
5	20
13	10
20	27
30	25
…	…



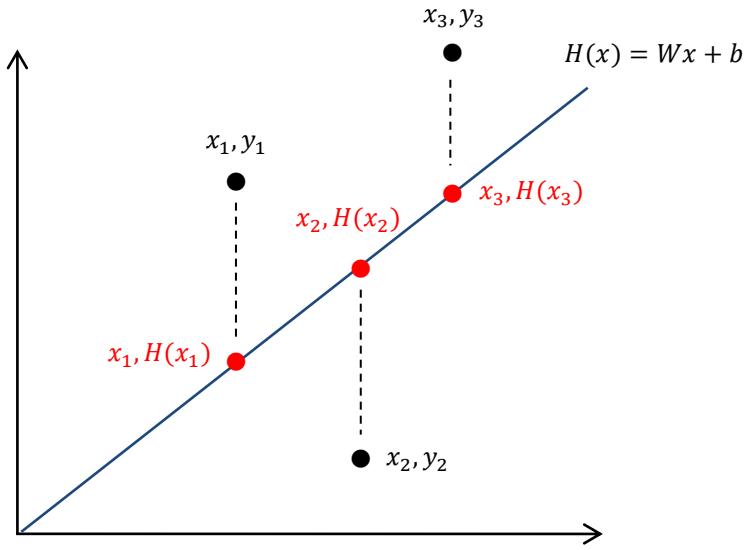
- 좌측과 같은 학습 데이터가 주어졌을 때, 이를 좌표 평면 위에 점으로 표시
- Linear regression 문제를 학습한다는 것은 학습 데이터의 모든 데이터 (좌표 평면 위의 모든 점)을 최대한 잘 표현할 수 있는 하나의 직선을 찾는 과정이라고 볼 수 있다.
- 학습 데이터를 좌표 평면 위에 점으로 표시했을 때, 모든 점들을 가장 잘 표현하기 위해 직선이 사용될 수 있다면, 이 학습 데이터는 Linear regression을 통해 문제를 해결할 수 있다고 가정한다.
- 때문에, 어떤 기계 학습 방법을 사용할지 결정하기 전에, '학습 데이터와 해결하려는 문제'의 파악 과정이 우선되어야 한다.
- 학습을 통해 최적의 직선을 찾아냈다면, 임의의 입력에 대하여 정답을 예측할 수 있다. 그렇다면 그 직선은 어떻게 찾을 수 있을까?

3-2-1. 기계 학습 과정 예제 – Linear Regression (2/17)

- 가설 (Hypothesis)
  - 수학적으로 모든 직선은 다음의 수식으로 표현할 수 있다.
    - 직선의 방정식 : $y = ax + b$
  - Linear regression은 직선을 구하는 문제이므로, 아래의 수식을 가설로 사용한다.
    → y = $H(x) = Wx + b$
  - 학습 데이터에서 x, y는 주어지므로, W와 b를 찾는 문제로 생각할 수 있다.
    그렇다면, W와 b는 어떻게 찾을 것인가?
  - 최적의 $W\hat{}$와 $b\hat{}$를 갖는 가설 $H(x) = W\hat{}x + b\hat{}$는 모든 학습 데이터(좌표 평면 위의 모든 점)들과 가설(직선) 사이의 거리가 가장 짧은 경우이다.
    - 거리가 짧다는 것은 학습 데이터가 가설(직선) 근처에 있다는 뜻이다.
  - 무수히 많은 직선들 중에서 좌표 평면 위의 모든 점들 간의 거리가 가장 짧은 직선이 학습 데이터를 가장 잘 표현하는 직선이며, 이 직선을 자동으로 찾아내는 과정을 기계 학습이라고 한다.
    - 이 때, 거리를 계산하기 위해 사용되는 함수가 Loss function 또는 Cost function이다.

3-2-1. 기계 학습 과정 예제 – Linear Regression (3/17)

- 비용 함수 (Cost function)
  - Hypothesis
    - y = $H(x) = Wx + b$
  - 우측의 그래프에서 단순하게 거리를 계산하면,
    distance = $(y_1 - H(x_1)) + (y_2 - H(x_2)) + (y_3 - H(x_3))$
  - 모든 점들과 거리를 계산하여 가장 짧은 경우를 찾아야 하므로, 거리(오차)의 평균을 계산하여 평균이 가장 작은 경우의 직선을 찾는다.
  - 평균 오차를 구하기 위해 단순히 합하고 나눈다면, 부호가 다른 경우에 정보량 손실이 발생하기 때문에 각 오차를 제곱해서 더한 뒤 평균을 계산한다.
    - 절댓값을 쓰지 않고 제곱하는 이유는 거리가 클 수록 제곱하면 값이 훨씬 커지기 때문에, 거리가 큰 경우에 대하여 가중치를 부여하기 위함



$$cost(H(x)) = \frac{1}{m}\sum_{i=1}^{m}(H(x_i) - y_i)^2$$

3-2-1. 기계 학습 과정 예제 – Linear Regression (4/17)

- 최적화 (Optimize)
  - 목적 함수(Objective function)를 이용하여, 최댓값 또는 최솟값을 구하는 과정
    - Linear regression에서는 Objective function과 Cost function이 동일

$$cost(H(x)) = \frac{1}{m}\sum_{i=1}^{m}(H(x_i) - y_i)^2 \quad = \frac{1}{m}\sum_{i=1}^{m}((Wx_i + b) - y_i)^2 = cost(W, b)$$

  - 최종적으로 $cost(W, b)$의 값이 최소가 되는 $\boldsymbol{W}$와 $\boldsymbol{b}$를 구하는 문제

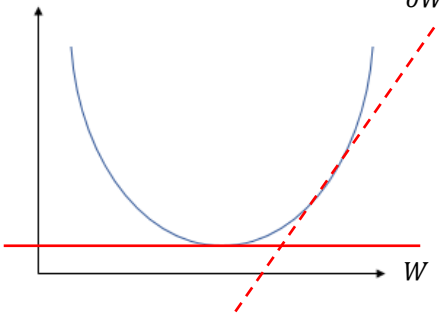
- ① 학습 데이터를 잘 표현할 수 있는 가설(Hypothesis 또는 관계식)을 정의 → Hypothesis : $H(x) = Wx + b$
- ② 가설을 이용하여 예측한 값과 실제 정답 사이의 오차를 계산 → Loss function
- ③ 모든 학습 데이터에 대하여, Loss function으로 오차를 구하고 오차의 평균을 계산 → Cost function
- ④ Cost function의 값(오차의 평균)이 최소가 되는 W와 b를 탐색 → Optimize

여기서, 평균 오차가 작다는 것은 모든 학습 데이터에 대하여 가설을 이용한 예측 값과 실제 정답이 비슷하다는 뜻이다.
정의한 가설($Wx + b$)에서 발생할 수 있는 무수히 많은 직선 중에, (W와 b의 값은 무수히 많으므로)
$cost(W, b)$의 값이 최소가 되는 W와 b를 찾는 과정을 최적화 과정이라고 한다.

3-2-1. 기계 학습 과정 예제 – Linear Regression (5/17)

- 최적화 (Optimize)
  - 그렇다면, 어떻게 $cost(W, b)$를 최소화할 것인가?
    Linear regression의 Cost function은 제곱하여 평균을 구하므로, 함수는 기울기가 양수인 2차 방정식의 형태가 된다.
  - 목적(비용) 함수를 W에 대한 2차 방정식으로 가정했을 때, 최소화는 2차 방정식의 최솟값을 구하는 것과 동일하며, 최종적으로 미분한 1차 방정식의 값이 '0'인 지점의 W를 구하면 된다.
  - 만약 입력 데이터가 하나만 주어졌다면, 단순한 대입 연산을 통해 W를 쉽게 찾을 수 있다.
    - W에 대한 2차 방정식으로 가정했기 때문에, W에 대하여 편미분(∂)을 수행한다. (W를 제외한 x와 y는 일반 상수 취급)
      → 즉, 하나의 입력이 주어진다면 하나의 2차 방정식을 푸는 것과 동일하다.

$$Z = (Wx - y)^2 \qquad \frac{\partial}{\partial W}(Wx - y)^2 = 2(Wx - y)x$$



(x, y)	$Wx - y = 0$	$\therefore W$
(28, 92)	$28W - 92 = 0$	$\frac{92}{28}$

(x, y)	$2(Wx - y)x = 0$	$\therefore W$
(28, 92)	$2(28W - 92)28 = 0$	$\frac{92}{28}$

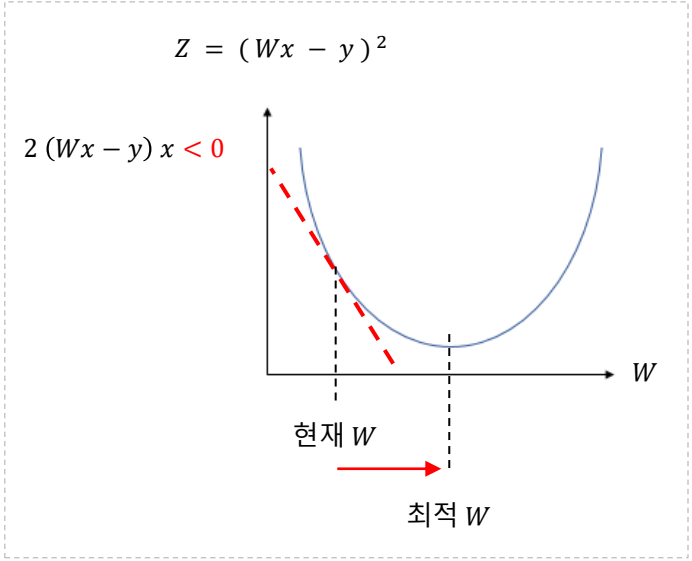
3-2-1. 기계 학습 과정 예제 – Linear Regression (6/17)

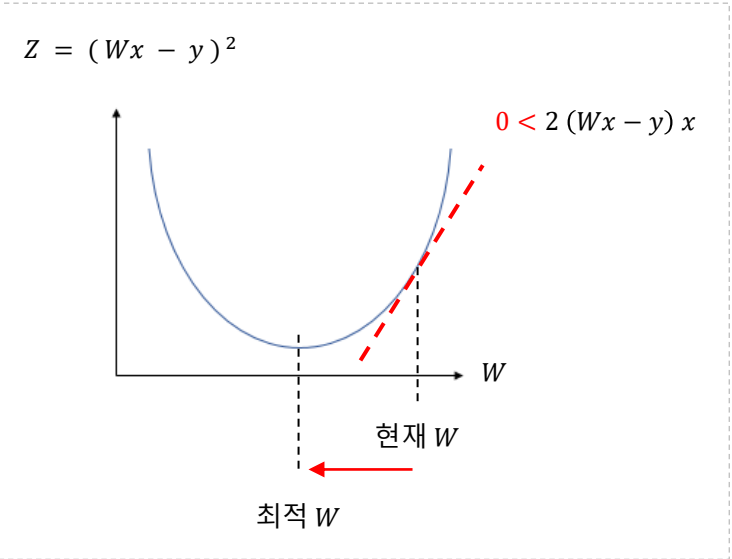
- 최적화 (Optimize)
  - 하지만, 기계 학습은 하나의 데이터가 아닌 수많은 데이터에 대하여 하나의 최적 W를 찾는 과정이다.

    - 먼저, 하나의 2차 방정식에서 바로 대입하지 않고, 최적의 W를 찾는다면, 어떤 원리일까?
      - 특정 W에서 기울기를 계산했을 때, 기울기가 음수라면 W는 오른쪽으로 이동해야 한다.
        특정 W에서 기울기를 계산했을 때, 기울기가 양수라면 W는 왼쪽으로 이동해야 한다.

        (우리가 구하려는 W가 변수이고 x, y는 데이터에서 주어지는 상수이다.)

$$\frac{\partial}{\partial W}(Wx-y)^2 = 2(Wx-y)x$$





3-2-1. 기계 학습 과정 예제 – Linear Regression (7/17)

- 최적화 (Optimize)
  - 주어진 모든 데이터에서 각 데이터 별로 2차 방정식이 만들어진다면, 데이터의 수만큼 2차 방정식이 생성되고, 모든 2차 방정식에서 최소화를 만족하는 하나의 최적 W를 찾아야 한다.

    - 모든 데이터에 대하여, 단순 대입 연산으로 하나의 최적 W를 찾는 것이 불가능하다.
      (아래 표에서 각 2차 방정식의 최소값을 만족하는 W는 전부 다른 것을 확인할 수 있다.)

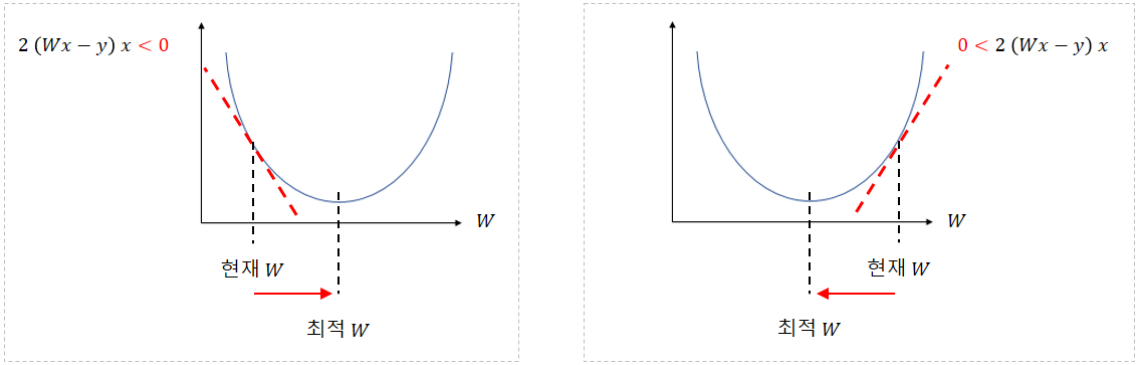
(x, y)	$2(Wx - y)x = 0$	$\therefore W$
(28, 92)	$2(28W - 92)28 = 0$	$\frac{92}{28}$
(21, 70)	$2(21W - 70)21 = 0$	$\frac{70}{21}$
(17, 56)	$2(17W - 56)17 = 0$	$\frac{56}{17}$

    - 당연히 데이터가 많아질수록 대입 연산으로 하나의 최적 W를 찾는 것은 더욱 더 어려워지며,
      결국, 데이터로부터 생성되는 모든 2차 방정식들을 모두 최소값으로 근사 시킬 수 있는 하나의 최적 W를 찾아야 한다.

3-2-1. 기계 학습 과정 예제 – Linear Regression (8/17)

- 최적화 (Optimize)
  - 그렇다면, 모든 2차 방정식들을 최소값으로 근사 시킬 수 있는 하나의 최적 W는 어떻게 구할 수 있을까?

    - 2차 방정식에서 최적 W를 구하는 방법은 현재 W에서의 기울기를 계산하고, 기울기의 부호에 따라 W를 이동하면 된다고 했다.

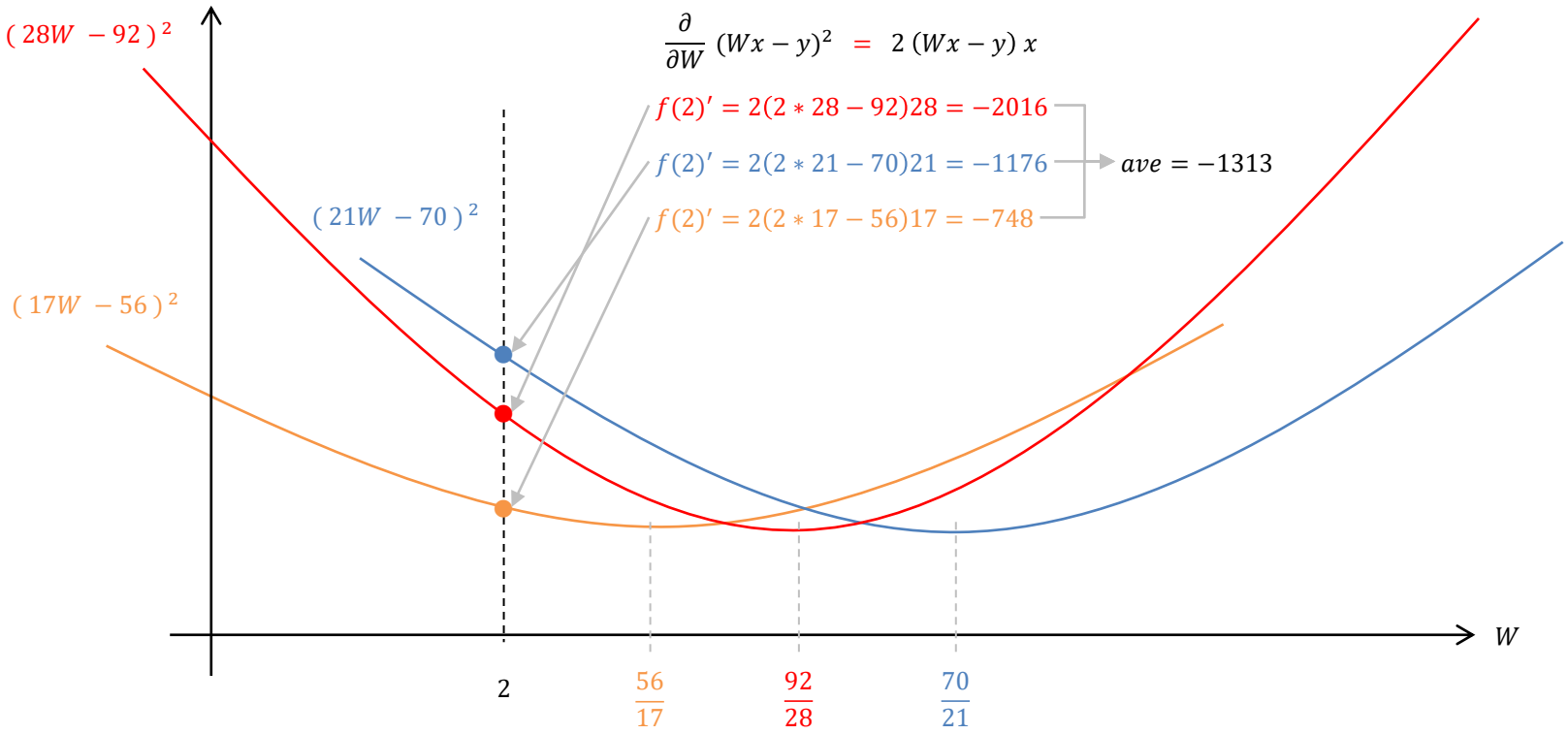


    - 모든 2차 방정식에서 하나의 최적 W를 구하려면, 현재의 W에서 모든 2차 방정식의 기울기를 계산하고, 그 기울기의 평균을 이용하여 W를 갱신하는 방법을 사용한다.

$$W := W - \alpha \frac{1}{m} \sum_{i=1}^{m} (Wx_i - y_i)\, x_i$$

3-2-1. 기계 학습 과정 예제 – Linear Regression (9/17)

- 최적화 (Optimize)
  - 먼저, 현재 W에서 모든 2차 방정식의 기울기를 구하고, 그 기울기의 평균을 계산



3-2-1. 기계 학습 과정 예제 – Linear Regression (10/17)

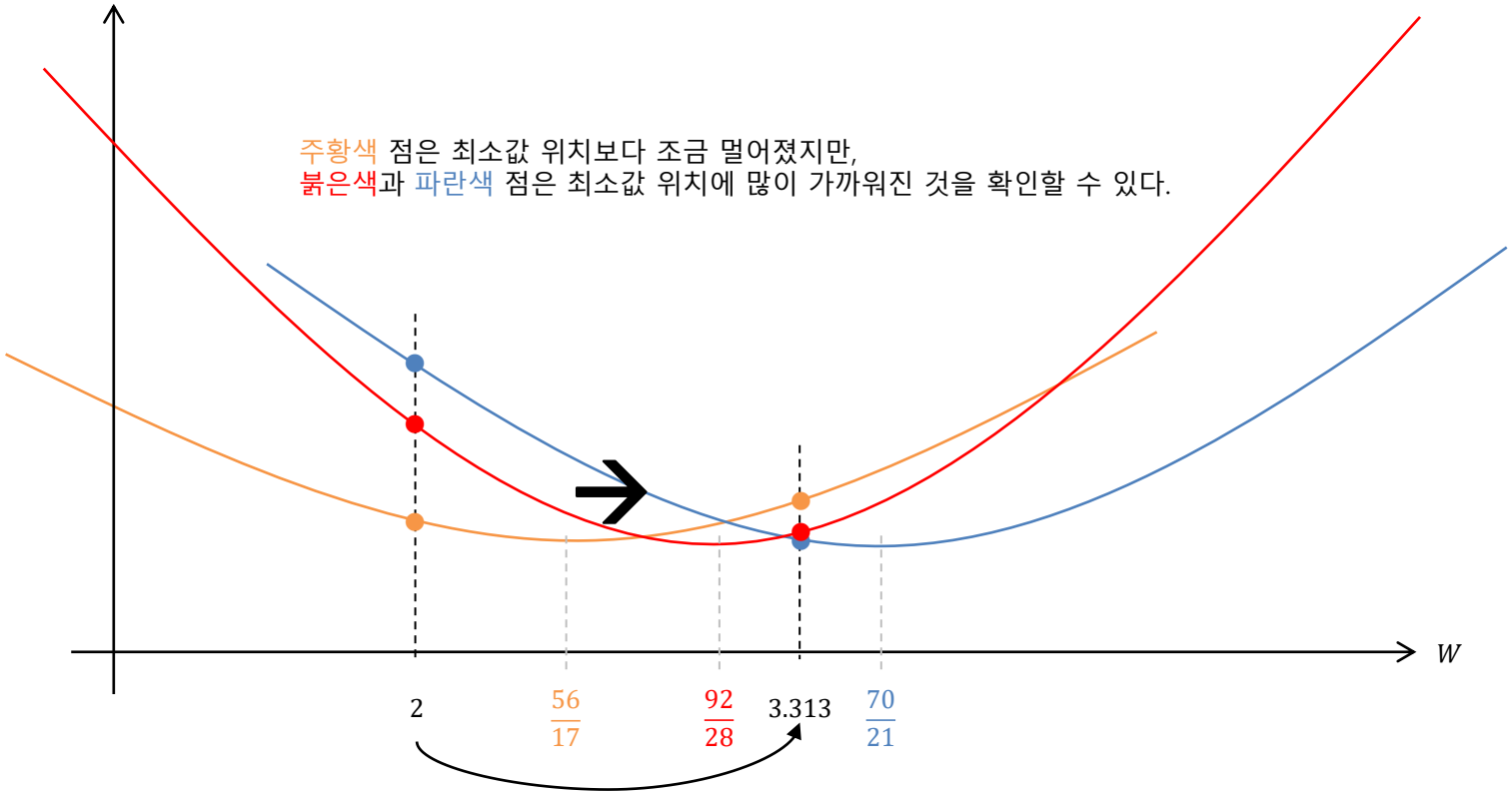
- 최적화 (Optimize)
  - 현재 W에서 계산된 평균 기울기를 이용하여, $Cost$가 최소가 되는 W로 업데이트
    - 현재 W에서 계산한 평균 기울기에 학습률(α: Learning rate)을 곱하여, W를 업데이트한다.
    - W는 기울기가 음수이면 증가, 양수이면 감소해야 하므로, 기존 W에서 계산된 값을 빼주면 된다.
    - 예제에서 계산된 평균 기울기에 학습률(0.001)을 곱한 값은 -1.313이고, 기존 값 2에서 빼주면 3.313으로 W를 갱신할 수 있다.

$$W := W - \alpha \frac{1}{m} \sum_{i=1}^{m} (Wx_i - y_i) x_i$$

$$W := 3.313 = 2 - (0.001 * -1313)$$

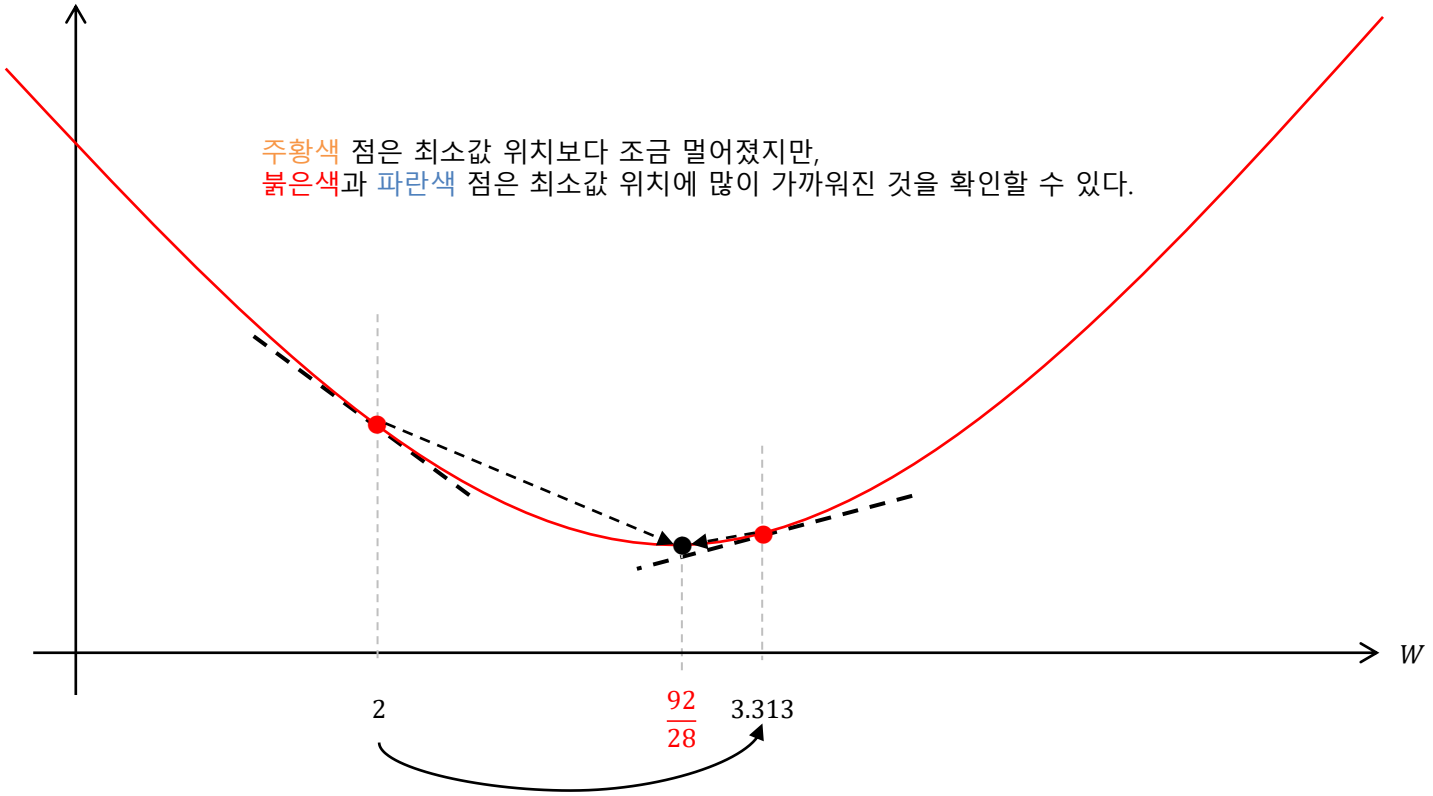
3-2-1. 기계 학습 과정 예제 – Linear Regression (11/17)

- 최적화 (Optimize)
  - 그렇다면, 모든 2차 방정식들의 최소값으로 근사할 수 있는 하나의 최적 w는 어떻게 구할 수 있을까?



3-2-1. 기계 학습 과정 예제 – Linear Regression (12/17)

- 최적화 (Optimize)
  - 그렇다면, 모든 2차 방정식들의 최소값으로 근사할 수 있는 하나의 최적 w는 어떻게 구할 수 있을까?



3-2-1. 기계 학습 과정 예제 – Linear Regression (13/17)

- 최적화 (Optimize)
  - 그렇다면, 모든 2차 방정식들의 최소값으로 근사할 수 있는 하나의 최적 W는 어떻게 구할 수 있을까?

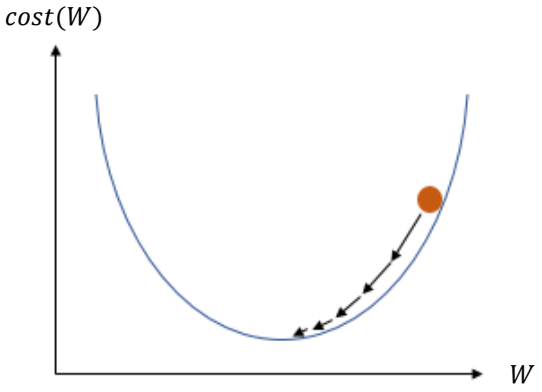
    - 학습이 진행될수록, 평균 오차가 줄어들고, 정답에 가까워지는 것을 확인할 수 있다.
      (모든 데이터에 대해서, 평균 오차가 '0'이 되는 것은 매우 어렵기 때문에, 학습이 진행되면서 오히려 오차가 늘어나는 경우도 발생한다.)

x	W	$y' = Wx$	y	$loss = (y' - y)^2$	$\leftarrow ave$	$\frac{\partial}{\partial W}(y' - y)^2 = 2(Wx - y)x$	$\leftarrow ave$	$ave * \alpha$
28	2	56	92	1296	854.66	-2016	-1313.33	-1.31333
21		42	70	784		-1176		
17		34	56	484		-748		

x	W	$y' = Wx$	y	$loss = (y' - y)^2$	$\leftarrow ave$	$\frac{\partial}{\partial W}(y' - y)^2 = 2(Wx - y)x$	$\leftarrow ave$	$ave * \alpha$
28	3.313	92.77	92	0.60	0.29	43.31	12.26	0.01226
21		69.58	70	0.18		-17.64		
17		56.33	56	0.11		11.11		

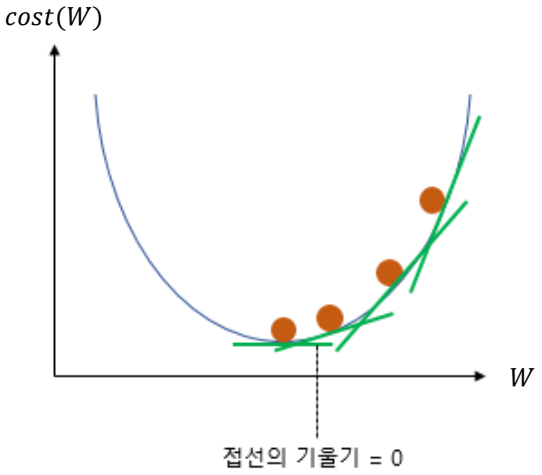
x	W	$y' = Wx$	y	$loss = (y' - y)^2$	$\leftarrow ave$	$\frac{\partial}{\partial W}(y' - y)^2 = 2(Wx - y)x$	$\leftarrow ave$	$ave * \alpha$
28	3.301	92.43	92	0.18	0.22	24.09	-0.11	-0.00011
21		69.32	70	0.46		-28.45		
17		56.12	56	0.01		4.02		

3-2-1. 기계 학습 과정 예제 – Linear Regression (14/17)



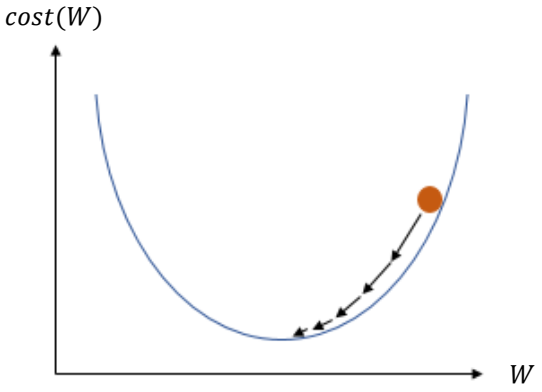
- 초기의 W는 임의의 값
- $cost(W)$가 최소가 되는 즉, 기울기가 '0'이 되는 W를 찾는 문제
  → 기울기가 '0'이 되도록 W의 값을 조정

$$W := W - \alpha \frac{1}{m} \sum_{i=1}^{m} (Wx_i - y_i) x_i$$



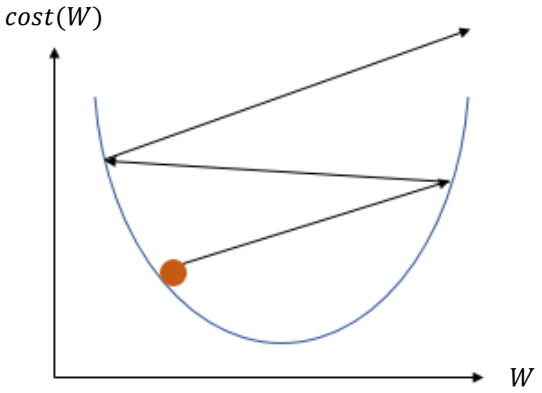
- 현재 W에서의 기울기가 '+'이면 위 수식에 의하여 W의 값은 감소하므로 왼쪽으로 이동하고, 이는 기울기가 '0'인 방향으로 W의 값을 조정한다.
- 현재 W에서의 기울기가 '-'이면 위 수식에 의하여 W의 값은 증가하므로 오른쪽으로 이동하고, 이는 기울기가 '0'인 방향으로 W의 값을 조정한다.
- W를 점점 변화시켜 기울기가 '0'인 위치로 유도하기 때문에 일반적으로 Linear regression의 최적화 알고리즘으로 경사 하강법(GDA: Gradient Descent Algorithm)을 사용한다.

3-2-1. 기계 학습 과정 예제 – Linear Regression (15/17)



$$W := W - \alpha \frac{1}{m} \sum_{i=1}^{m} (Wx_i - y_i)\, x_i$$

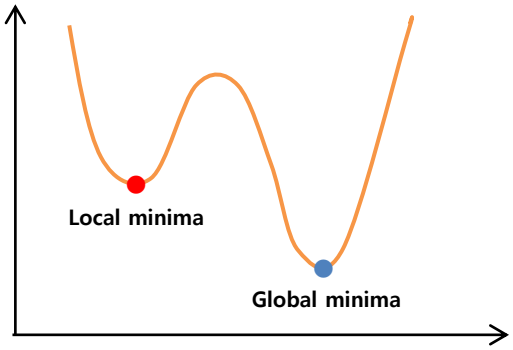
- 학습률 $(\alpha: Learning\ rate)$
  - W의 값을 어느 정도로 변경할 것인가?

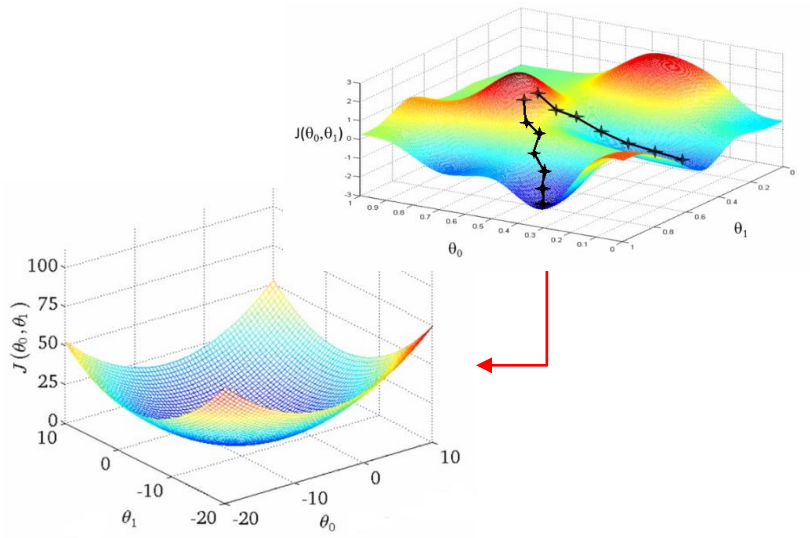


  - 또는 W를 그래프의 한 점으로 봤을 때,
    접선의 기울기가 '0'이 될 때까지 경사를 따라 내려간다는 관점에서
    얼마나 큰 폭으로 이동시킬 것인가?
  - α의 값이 너무 작으면, 접선의 기울기가 '0'인 W까지 이동하는데
    너무 오랜 시간이 걸린다.
  - α의 값이 너무 크면, 접선의 기울기가 '0'인 W를 지나쳐서
    좌우로 발산하게 된다.

3-2-1. 기계 학습 과정 예제 – Linear Regression (16/17)

- 지역 최솟값 문제 (Local Minima Problem)
  - 초기 W 값에 따라 잘못된 최솟값으로 수렴하는 문제
    - 'Local minima'로 수렴 : 잘못 학습된 경우
    - 'Global minima'로 수렴 : 올바르게 학습된 경우
  - Objective(Cost) function이 Local minima 문제에 빠지지 않는지 확인
    - 초기 W 값에 상관 없이 하나의 최솟값으로 수렴한다면, Convex function이라고 부른다.
    - 즉, 함수를 설계할 때 Convex function 가능한지 반드시 확인해야 한다.





3-2-1. 기계 학습 과정 예제 – Linear Regression (17/17)

- 최종 정리
  - Linear Regression은 직선의 방정식을 가설로 사용한다.
  - 가설에 의해 무수히 많은 직선이 있을 수 있고,
    이 무수히 많은 직선들 중에서 학습 데이터를 가장 잘 표현할 수 있는 하나의 직선을 찾는 것이 목적
  - 모든 학습 데이터(좌표 평면 위에 있는 모든 점)들을 가장 잘 표현할 수 있는 직선이란?
    - 좌표 평면 위의 모든 점들과 가장 거리가 가까운 직선
  - 좌표 평면 위의 모든 점들과 가장 거리가 가깝다는 것은 각각의 점들과 직선 사이의 거리를 전부 계산했을 때,
    이 거리들을 제곱하여 평균을 구한 값이 최솟값을 가진다는 뜻
  - Linear Regression에서 모든 점들과 직선 사이의 거리를 계산하고 제곱하여 평균을 구하는 함수가 Cost function
  - Cost function의 값이 최소가 되는 직선은 학습 데이터를 가장 잘 표현하는 직선이며, 이 직선을 찾아내는 과정(최적화: Optimize)을 Linear regression 문제를 학습한다고 하며, 찾아낸 직선을 이용하여 임의의 입력 x값에 대한, 정답 y를 예측하는 것이 Machine learning의 목적이다.

3-2-2. 기계 학습 과정 예제 – Logistic Regression (Binary Classification) (1/11)

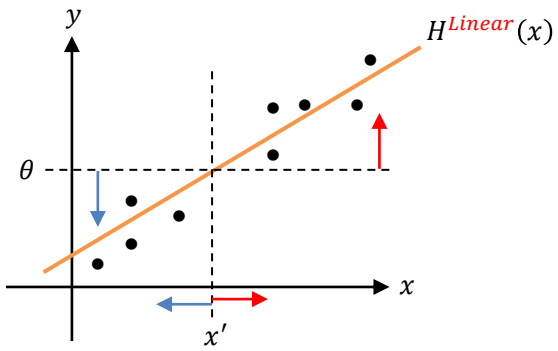
- 분류(Classification) 문제란?
  - 분류 문제는 연속된 값이 아니라, 여러 개의 값 중에서 하나로 예측하는 것을 말한다.
    - 분류 문제는 N개의 정답 중에서, 한 가지로 예측하는 것을 뜻하며, 일반적으로 분류 대상인 N개의 정답을 레이블(Label)이라고 표현한다.
      (Linear regression은 연속된 값을 예측하는데 사용)
  - 대표적인 분류 문제
    - 이진 분류 문제 (Binary classification)
      - 스팸 메일인가? 아닌가?
      - 긍정인가? 부정인가?
    - 다중 분류 문제 (Multi-label classification)
      - 대학교 학점(A, B, C, D, F) 예측
      - 리뷰 평점(1, 2, 3, 4, 5) 분류
      - 개체명(Person, Organization, Location, …) 분류
  - 그렇다면, 분류 문제를 풀기 위해서는 어떻게 해야 하는가?
    또, 분류 문제를 Linear regression으로 푸는 것은 불가능한가?

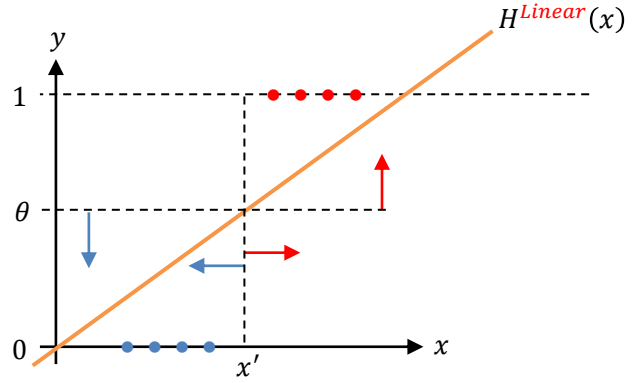
3-2-2. 기계 학습 과정 예제 – Logistic Regression (Binary Classification) (2/11)

- (이진) 분류 문제를 Linear Regression으로 푸는 것은 불가능한가?
  - 절대 불가능한 것은 아니다.
    - 만약 주어진 데이터가 매우 간단하다면, Linear regression을 이용하여 이진 분류 문제를 해결할 수도 있다.
      Linear regression은 연속된 값을 예측하지만, 예측된 값을 특정 임계치(θ)를 기준으로 나눌 수 있다면 이진 분류 문제 해결이 가능하다.

$$H^{Linear}(x) \leq \theta = 0$$
$$H^{Linear}(x) > \theta = 1$$

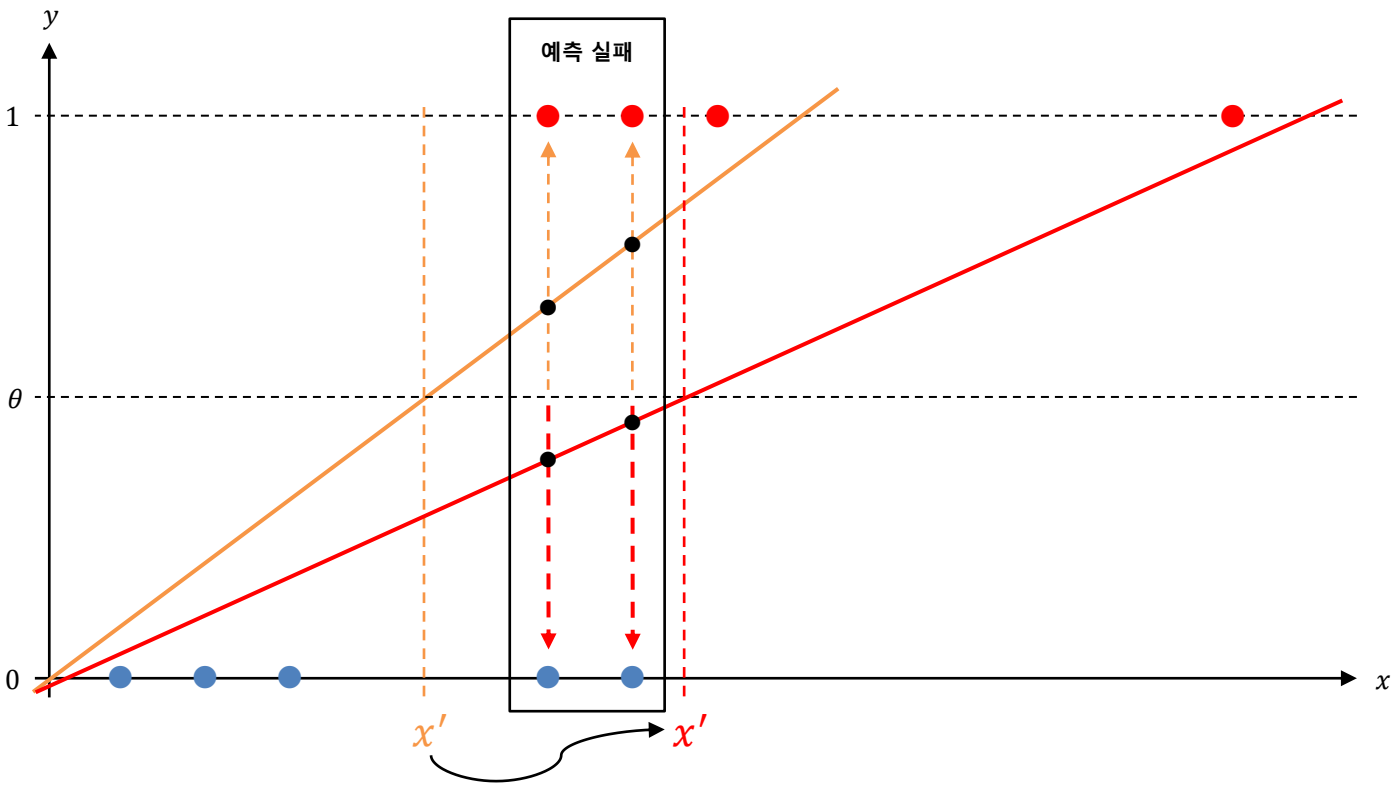
    - 위 수식처럼 임계치를 기준으로 '0'과 '1'로 나눈다면, 이진 분류 문제를 푸는 것이 가능해진다.
      - 임계치는 임의의 값이지만, 일반적으로 예측 범위의 가운데 지점을 사용한다.





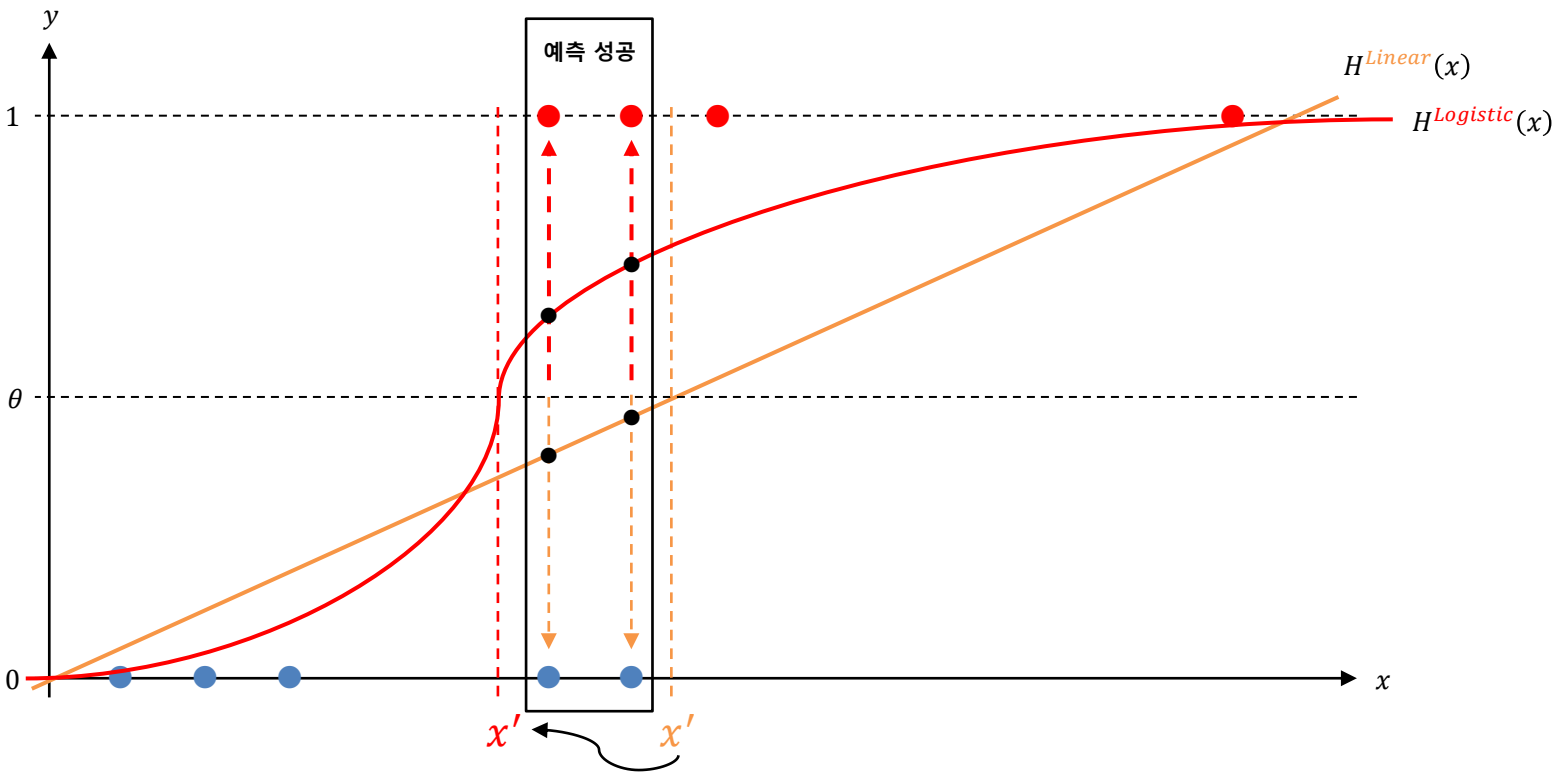
3-2-2. 기계 학습 과정 예제 – Logistic Regression (Binary Classification) (3/11)

- (이진) 분류 문제를 Linear Regression으로 푸는 것은 불가능한가?
  - 데이터가 간단하지 않은 경우라면, Linear regression으로 이진 분류 문제를 해결하는 것은 매우 어려워진다.



3-2-2. 기계 학습 과정 예제 – Logistic Regression (Binary Classification) (4/11)

- 그렇다면, (이진) 분류 문제를 어떻게 해결할 것인가?
  - 이진 분류 문제를 해결하기 위한 대표적인 알고리즘으로 Logistic regression이 있으며, 원리는 다음과 같다.



3-2-2. 기계 학습 과정 예제 – Logistic Regression (Binary Classification) (5/11)

- 이진 분류 문제 해결을 위한 Logistic Regression의 가설 (Hypothesis)
  - 이진 분류는 Linear regression처럼 연속된 값이 아닌, '0' 또는 '1'의 값을 예측하는 문제
    - 먼저, 이를 위해 예측 값의 범위를 0 ~ 1 사이의 값으로 고정
    - 고정된 0 ~ 1 사이의 값에서 0.5를 기준으로 이하이면 '0', 초과이면 '1'을 예측하여 이진 분류 문제를 해결
  - 즉, 연속된 실수 값을 0 ~ 1 사이의 값으로 정규화하기 위한, 새로운 가설 필요
    - $H^{Linear}(x) = Wx + b = z$일 때, $g(z)$가 0 ~ 1 사이의 값을 가지는 새로운 가설 $g(z) = g(H^{Linear}(x))$을 사용
  - $H^{Linear}(x) = Wx + b$의 실수 전체 범위를 0 ~ 1 사이의 값으로 정규화하는 함수 g를 Sigmoid 함수라고 부른다.

$$H^{Linear}(x) = Wx + b = z$$

$$H^{Logistic}(z) = Sigmoid(z) = \frac{1}{1 + e^{-z}} = \frac{1}{1 + e^{-(Wx+b)}}$$

3-2-2. 기계 학습 과정 예제 – Logistic Regression (Binary Classification) (6/11)

- 이진 분류 문제 해결을 위한 Logistic Regression의 비용 함수 (Loss/Cost function)
  - Logistic regression의 출력은 0 ~ 1 사이의 값이고, 정답(Label)은 '0' 또는 '1'이다.
    - 즉, 출력과 정답 사이의 오차를 계산하는 비용 함수는 다음 4가지 경우에 대하여 만족될 수 있어야 한다.

출력 ($H^{Logistic}(z) = Sigmoid(z)$)	정답(Label)	Loss 또는 Cost
'0'에 가까운 경우	0	최소
'1'에 가까운 경우	0	최대
'0'에 가까운 경우	1	최소
'1'에 가까운 경우	1	최대

- 위 4가지 경우를 만족하는 함수는 다음과 같이 정의될 수 있다.

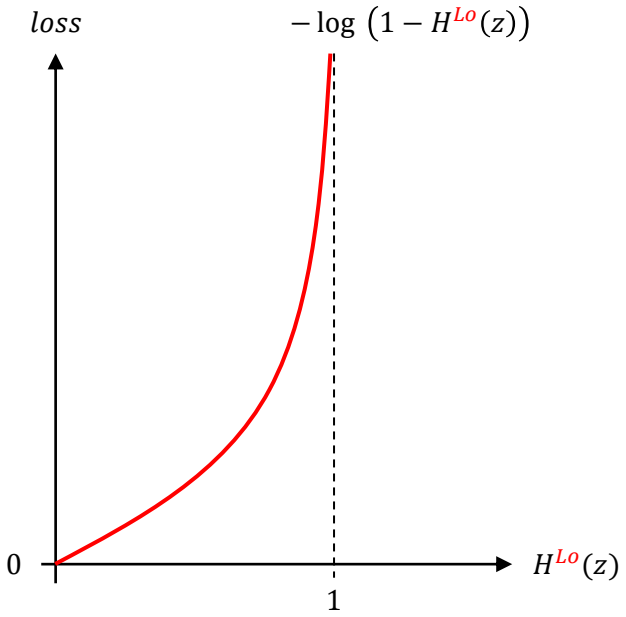
$$loss(H^{Lo}(z), y) = \begin{cases} -\log\left(1 - H^{Lo}(z)\right) & : y = 0 \\ -\log\left(H^{Lo}(z)\right) & : y = 1 \end{cases}$$

$$cost(H^{Lo}(z), y) = \frac{1}{m}\sum_{i=1}^{m} loss(H^{Lo}(z), y)$$

3-2-2. 기계 학습 과정 예제 – Logistic Regression (Binary Classification) (7/11)

- 이진 분류 문제 해결을 위한 Logistic Regression의 비용 함수 (Loss/Cost function)
  - 정답(Label)이 '0'인 경우, 출력이 '0'에 가깝다면 최소 오차, '1'에 가깝다면 최대 오차를 만족해야 한다.

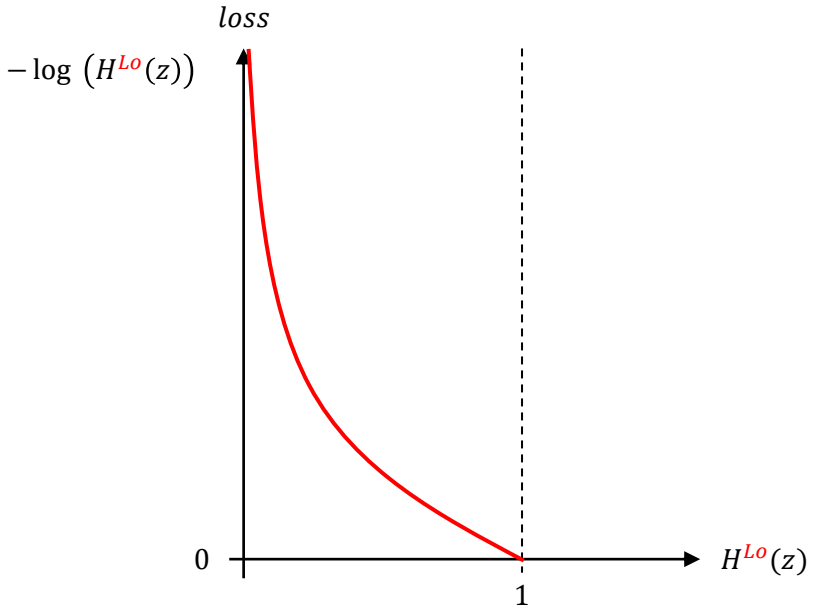
$$loss(H^{Lo}(z), y) = \begin{cases} -\log\left(1 - H^{Lo}(z)\right) & : y = 0 \\ -\log\left(H^{Lo}(z)\right) & : y = 1 \end{cases}$$



3-2-2. 기계 학습 과정 예제 – Logistic Regression (Binary Classification) (8/11)

- 이진 분류 문제 해결을 위한 Logistic Regression의 비용 함수 (Loss/Cost function)
  - 정답(Label)이 '1'인 경우, 출력이 '0'에 가깝다면 최대 오차, '1'에 가깝다면 최소 오차를 만족해야 한다.

$$loss(H^{Lo}(z), y) = \begin{cases} -\log\left(1 - H^{Lo}(z)\right) & : y = 0 \\ -\log\left(H^{Lo}(z)\right) & : y = 1 \end{cases}$$

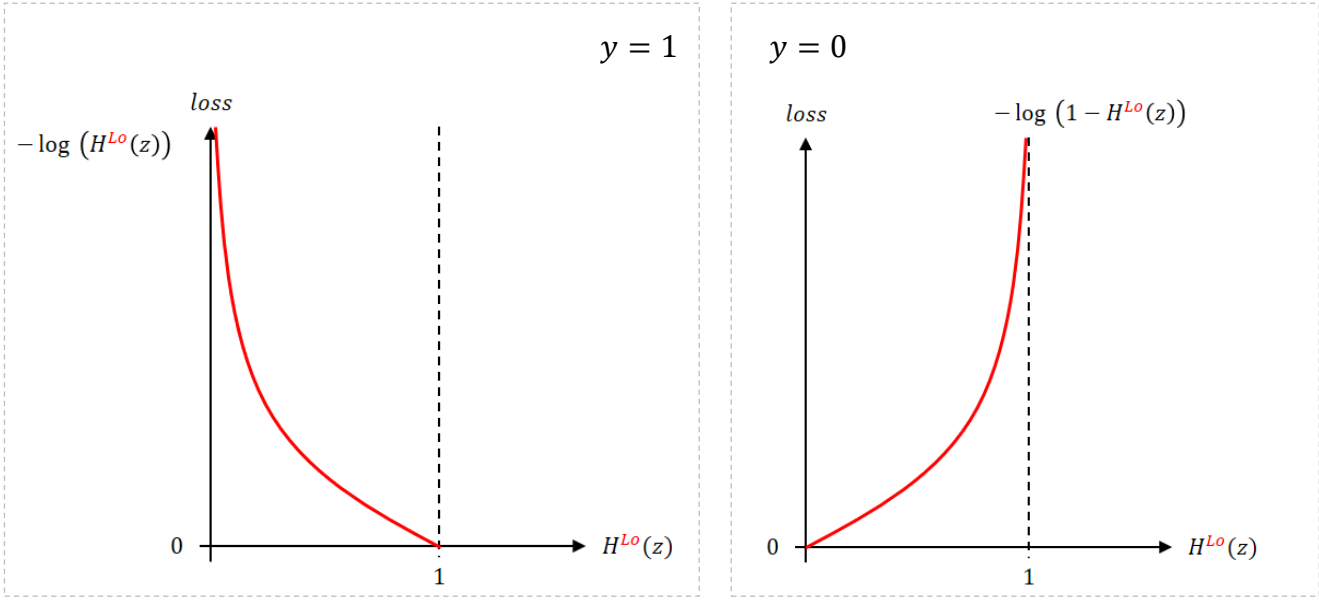


3-2-2. 기계 학습 과정 예제 – Logistic Regression (Binary Classification) (9/11)

- 이진 분류 문제 해결을 위한 Logistic Regression의 비용 함수 (Loss/Cost function)

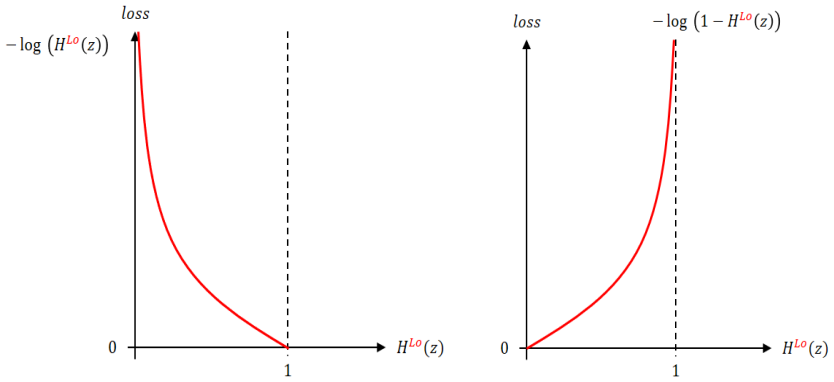
$$loss(H^{Lo}(z), y) = \begin{cases} -\log\left(1 - H^{Lo}(z)\right) & : y = 0 \\ -\log\left(H^{Lo}(z)\right) & : y = 1 \end{cases}$$

$$cost(H^{Lo}(z), y) = \frac{1}{m}\sum_{i=1}^{m} loss(H^{Lo}(z), y)$$



3-2-2. 기계 학습 과정 예제 – Logistic Regression (Binary Classification) (10/11)

- 이진 분류 문제 해결을 위한 Logistic Regression의 비용 함수 (Loss/Cost function)
  - 그래프를 보면 알 수 있듯이, Logistic regression의 Loss(Cost) function은 Convex function이고 W에 대하여 편미분한 수식을 이용하여, Linear regression과 동일하게 경사 하강법을 이용한 최적화를 수행할 수 있다.



  - 정답(Label)과 상관 없이 하나로 사용하기 위한 수식은 다음과 같다.

$$-y * \log\left(H^{Lo}(z)\right) - (1-y) * \log\left(1-H^{Lo}(z)\right)$$

3-2-2. 기계 학습 과정 예제 – Logistic Regression (Binary Classification) (11/11)

- 최종 정리
  - Logistic regression은 Sigmoid 함수를 가설로 사용한다.
    - Linear regression과 마찬가지로, 매개 변수에 대하여 편미분한 수식을 이용한 경사 하강법으로 최적화를 수행한다.

	Hypothesis	***Loss***	***Cost***	***Optimize***
Linear	$H^{Li}(x) = Wx + b$	$(H^{Li}(x) - y)^2$	$\frac{1}{m}\sum Loss$	$W := W - \alpha * \frac{\partial}{\partial W} Cost$
Logistic	$H^{Lo}(z) = Sigmoid(z)$ $= \frac{1}{1+e^{-z}}$ $= \frac{1}{1+e^{-(Wx+b)}}$	$\begin{cases} -\log(1 - H^{Lo}(z)) & : y = 0 \\ -\log(H^{Lo}(z)) & : y = 1 \end{cases}$ $= -y * \log(H^{Lo}(z)) - (1-y) * \log(1 - H^{Lo}(z))$	$\frac{1}{m}\sum Loss$	$W := W - \alpha * \frac{\partial}{\partial W} Cost$

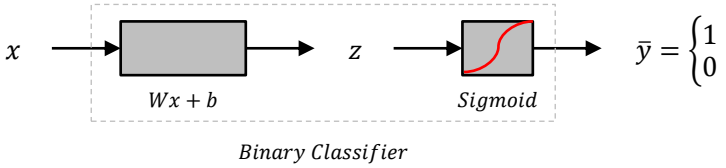
  - 기울기 베이스의 학습을 수행하는 모델의 Cost function과 Optimize의 일반적인 개념은 동일하다.
    - 가설에 대한 비용(오차) 함수를 정의하고, 비용 함수를 매개 변수에 대해 편미분한 수식으로 최적화(Optimize) 수행

3-2-3. 기계 학습 과정 예제 – 다중 분류 문제 (Multi-label Classification) (1/5)

- 다중 분류 문제(Multi-label Classification)이란?
  - N개의 정답(Label) 중에서 한 가지로 예측하는 문제이며, 일반적으로 다음 두 가지 방법을 이용하여 문제를 해결한다.

    - ① 여러 개의 이진 분류기 사용
    - ② Softmax 함수 사용

      - 먼저, Sigmoid 함수를 가설로 사용하는 Logistic regression을 이용한 이진 분류기(Binary classifier)는 다음과 같다.



- Logistic regression을 이용한 이진 분류기의 개념을 다시 정리하면,

  (1) 전체 데이터를 가장 잘 표현할 수 있는 하나의 직선을 찾는다.

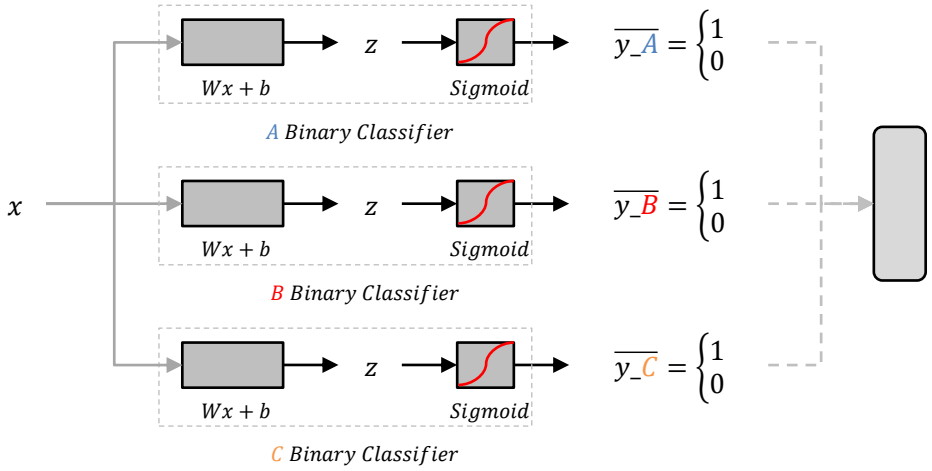
  (2) 직선의 출력 범위를 0 ~ 1 사이로 고정하기 위해 Sigmoid 함수를 사용하여 정규화한다.

  (3) Sigmoid 함수의 출력 값이 '0.5'를 기준으로 이하이면 '0', 초과이면 '1'을 반환한다.

  (단, Sigmoid 함수를 사용하지 않은 Linear regression인 경우에는 '0.5'가 아닌 임의의 값 θ를 기준으로 이하이면 '0', 초과이면 '1'을 반환한다.)

3-2-3. 기계 학습 과정 예제 – 다중 분류 문제 (Multi-label Classification) (2/5)

- ① 여러 개의 이진 분류기를 사용한 다중 분류 문제 해결
  - 정답(Label)의 수가 N개라면, 동일하게 N개의 이진 분류기를 사용하면 다중 분류 문제를 해결할 수도 있다.
    - 각각의 분류기는 한 가지 정답인 경우에만 '1', 나머지 정답은 모두 '0'으로 치환하여 각각 학습을 진행한다.
      - 예를 들어, 3개의 정답(A, B, C)을 가지는 데이터라면, 3개의 이진 분류기를 학습해야하고,
        A를 분류하는 이진 분류기라면 정답이 A인 경우에만 '1', 나머지 B와 C인 경우에는 '0'으로 정답을 치환하여 이진 분류 학습을 진행한다.
    - 이렇게 학습된 이진 분류기들은 하나의 입력에 대하여, 각각의 이진 분류 예측 값을 반환하고, 사용자는 이를 토대로 최종 분류를 수행한다.
      - 하지만, 두 개 이상의 분류기에서 '1'을 예측하는 경우, 하나의 정답으로 최종 분류하는 것이 어려워지기 때문에 개념적으로만 이해할 뿐,
        실제로 사용되는 경우는 적은 편이다.



3-2-3. 기계 학습 과정 예제 – 다중 분류 문제 (Multi-label Classification) (3/5)

- ② Softmax 함수를 이용한 다중 분류 문제 해결
  - Softmax 함수는 출력 값이 0 ~ 1 사이의 값으로 정규화되며, 모든 출력 값의 총합은 항상 '1'이 된다.
    - Softmax는 지수 함수이기 때문에, 입력 값들의 대소 관계는 변하지 않는다.
      - 즉, 선형 값 또는 ReLU, Sigmoid 등의 다른 활성화 함수 값으로도 Argmax 함수를 이용하면, 가장 큰 값을 찾는 경우에는 동일하다.
  - 그렇다면, 왜(언제) Softmax 함수를 사용할까?

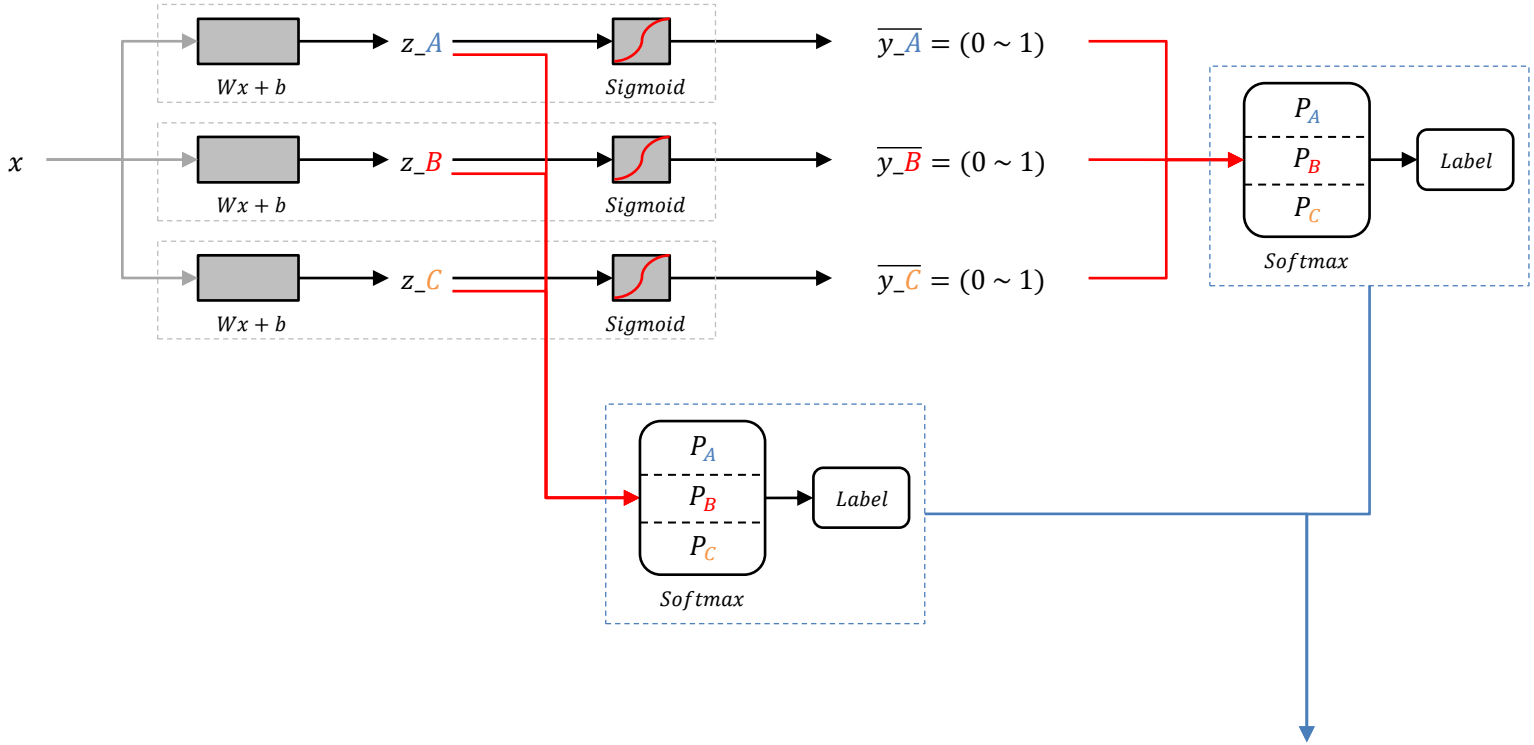
① 모든 출력 값의 총합이 '1'이 되므로, 확률 값으로 사용할 수 있다.
② 지수 함수이므로, 입력 값 중 큰 값은 더 크게, 작은 값은 더 작게 만들어서 출력 값이 더 잘 구분된다.
③ 지수 함수이므로, 미분하는 경우 자기 자신이 되고, 이는 미분을 이용한 계산 과정에서 편리하다.

- Linear regression에서 Sigmoid 함수를 이용한 정규화는 하나의 값에 대한 정규화 과정이다.
  (하나의 입력에 대한 출력 값의 범위를 실수 전체에서 0 ~ 1 사이로 정규화)

- Softmax 함수를 이용한 정규화는 하나의 값이 아닌, 여러 값에 대한 정규화 과정이다.
  (각각의 출력 값은 0 ~ 1 사이의 값을 가지며, 모든 출력 값의 총합은 반드시 '1'이다.)

3-2-3. 기계 학습 과정 예제 – 다중 분류 문제 (Multi-label Classification) (4/5)

- ② Softmax 함수를 이용한 다중 분류 문제 해결



① Softmax 함수 내에서 계산된 확률의 값은 다르지만, 대소 관계는 동일하다.

② Softmax 함수를 통해 출력된 정답(Label)은 동일하다.

3-2-3. 기계 학습 과정 예제 – 다중 분류 문제 (Multi-label Classification) (5/5)

- 최종 정리
  - 다중 분류 문제(Multi-label classification)은 *N*개의 정답(Label) 중에서, 한 가지로 예측하는 문제
    - 여러 개의 이진 분류기를 이용하여 문제를 해결할 수도 있지만, 일반적으로 Softmax 함수를 사용한다.
    - Softmax 함수는 단일 입력에 대하여 임계치를 기준으로 분류하는 것이 아닌, 다중 입력에 대한 정규화 함수이다.
      - 각 입력의 출력 값은 0 ~ 1 사이의 값이며, 모든 출력 값의 합은 반드시 '1'이 된다.
      - 다중 입력에 대한 대소 관계는 출력에서 변하지 않는다.

    - 다중 입력에 대하여, 최대 값을 가지는 하나의 정답을 찾는 문제라면, 일반적인 Argmax 함수를 이용해도 상관은 없다.
      - Argmax 함수는 단순히 가장 큰 값을 찾는 함수이므로, 함께 입력된 데이터 간의 상대적인 평가는 고려되지 않는다.
      - Softmax 함수는 모든 데이터의 출력 총합이 반드시 '1'이 되므로, 데이터 간의 상대적인 평가도 반영된다. → 확률로 사용 가능

    - Linear regression과 Sigmoid 함수를 사용한 Logistic regression, 그리고 Sigmoid 함수로부터 유도된 Softmax 함수의 특성은 다음과 같다.

      - ① Linear regression의 실수 전체 범위의 출력을 0 ~ 1 사이로 고정하는 역할이 Sigmoid 함수이다.
      - ② Linear regression에 Sigmoid 함수를 적용하여, 0 ~ 1 사이의 값을 '0.5'를 기준으로 '0' 또는 '1'로 분리하는 것이 Logistic regression이다.
      - ③ Linear regression과 Logistic regression은 단일 입력에 대한 처리를 수행한다.
      - ④ Softmax 함수는 여러 개의 입력을 받고, 각 출력 값은 0 ~ 1 사이의 값을 가지며, 모든 출력 값의 합은 반드시 '1'이 된다.
      - ⑤ Softmax 함수의 각 출력 값은 확률로 사용이 가능하며, 가장 큰 확률 값을 가지는 정답을 선택함으로써 다중 분류 문제를 해결할 수 있다.
      - ⑥ 가장 큰 값(확률)을 가지는 하나의 정답을 찾는 문제라면, Softmax 또는 Argmax 어떤 함수를 사용해도 상관없다.
      - ⑦ Argmax 보다 Softmax 함수를 사용하는 이유는 전체 출력 값의 합이 '1'이 되므로, 함께 입력된 데이터 간의 상대적인 평가가 반영된다.
      - ⑧ Linear regression 또는 Logistic regressio의 출력 값은 둘 다 Softmax 함수의 입력으로 사용 가능하고 Softmax의 출력은 동일하다.
      - ⑨ Linear regression과 Sigmoid를 통해 정규화된 Logistic regression의 출력 값을 입력으로 사용한 Softmax 함수 내부의 확률 값은 다를 수 있다.