1주차_강의 요약

태그	euron

AI: new Electricity→ 큰 변화를 보여주고 있다.

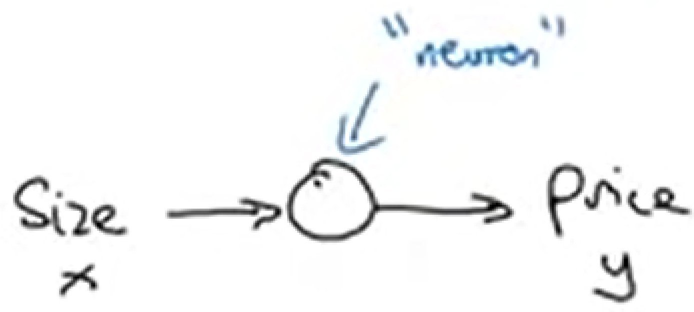
- ▼ 신경망이란 무엇인가.

  딥러닝: 신경망을 학습시키는 것

  신경망이란?

  ex 주택 가격 예측 예제

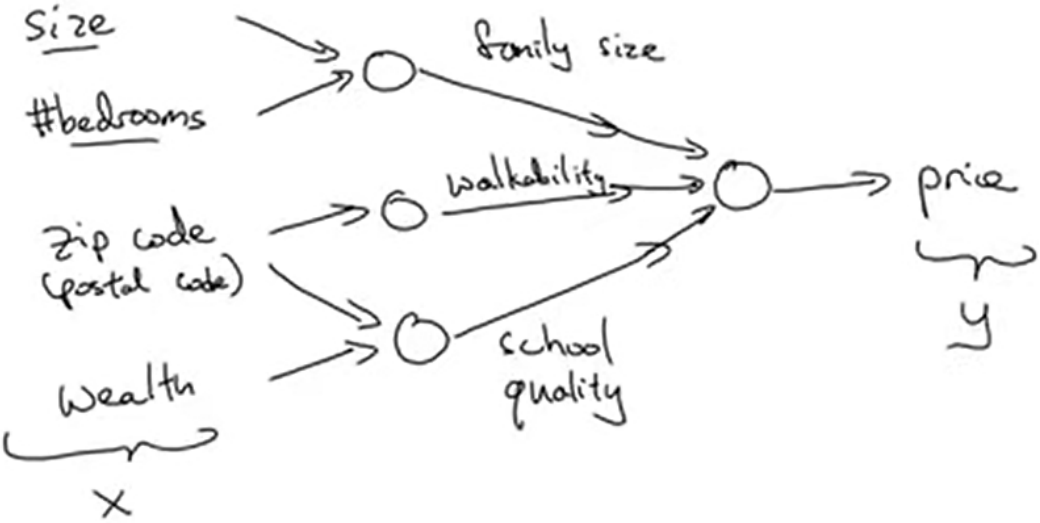
  주택 크기→주택 가격



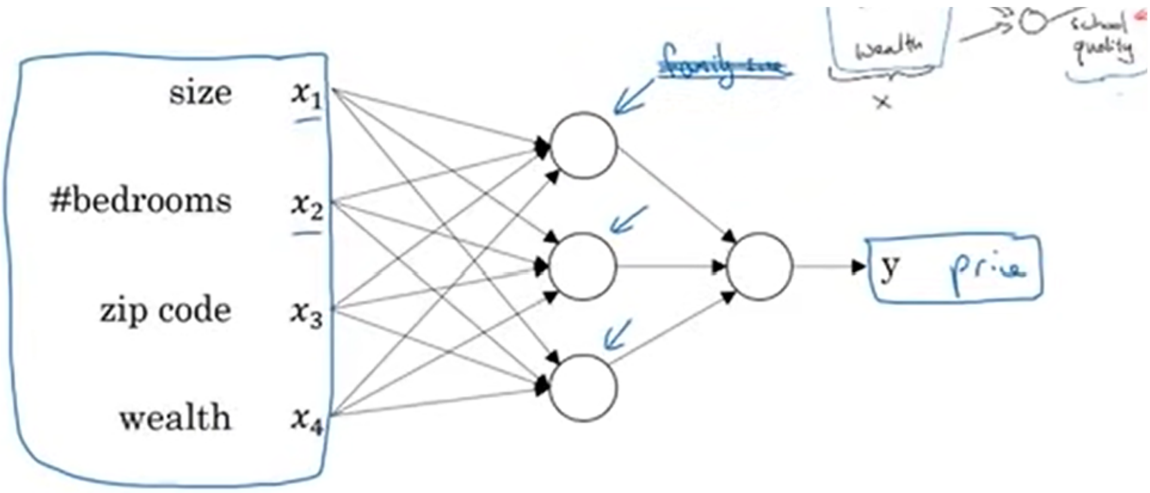
뉴런이 하는 일: 주택의 크기를 입력으로 받아서 선형 함수를 계산하고 함수의 값과 0 중 큰 값을 주택 가격으로 예측

ReLU: Rectified Linear Unit

작은 신경망 을 레고를 쌓는 것과 같이 신경망을 확장시킬 수 있음.



작은 원이 비선형 함수인 것. 뉴런이나 간단한 예측기들을 쌓아 올림으로써 이전보다 더 큰 신경망을 가지게 됨.



각 원들은 신경망의 은닉 유닛이라고 부름. 4개의 입력을 받음.

ex) 첫번째 노드가 가족의 크기를 내포한다고 하면, 어떤 계산을 하고 싶든지 4개의 입력을 다 받음. _Q. 이런 언급을 하는 이유는?

데이터의 양이 충분할 때, 신경망은 x를 y로 연결하는 함수를 알아내는 데 뛰어남.

Q. 결국 신경망의 핵심은 무엇일까? 비선형으로 입력 x를 출력 y에 연결하는 일?

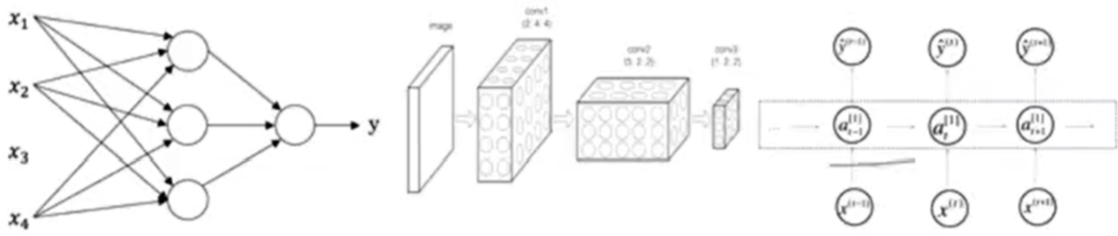
▼ 신경망을 이용한 지도학습

현재까지 신경망의 경제적인 가치도 머신러닝의 한 종류인 지도학습을 통해 계산됨.

Supervised Learning(지도학습) 적용 예시

Input(x)	Output (y)	Application	
Home features	Price	Real Estate	Standard NN
Ad, user info	Click on ad? (0/1)	Online Advertising	
Image	Object (1,…,1000)	Photo tagging	CNN
Audio	Text transcript	Speech recognition	RNN
English	Chinese	Machine translation	
Image, Radar info	Position of other cars	Autonomous driving	Custom/Hybrid

Neural Network examples



Standard NN　Convolutional NN　Recurrent NN

데이터의 종류

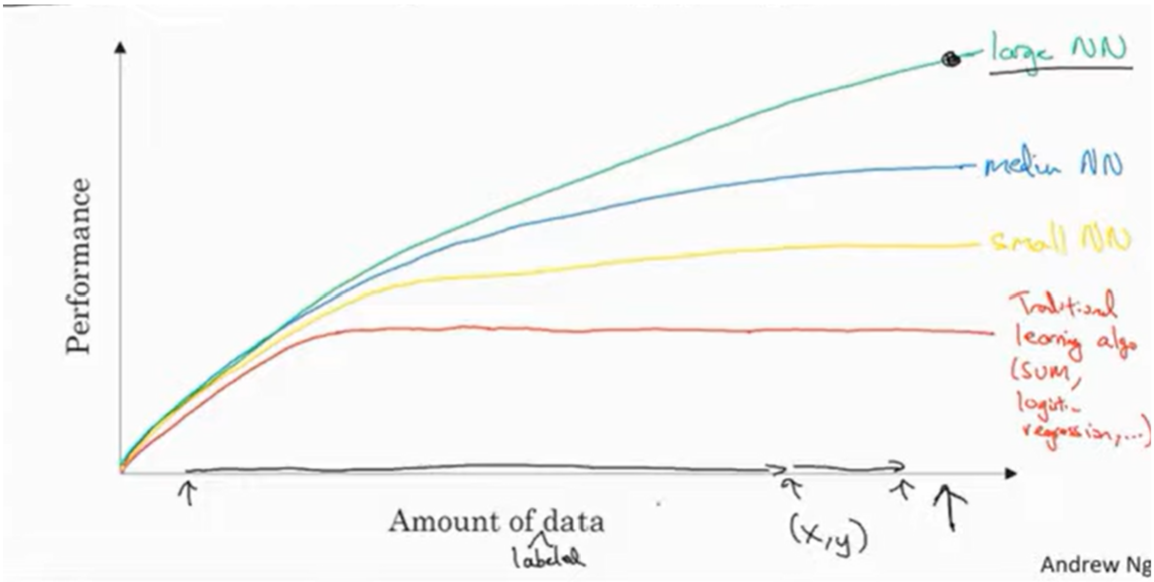
- 구조적 데이터: 데이터베이스로 표현된 데이터
- 비구조적 데이터: ex 오디오, 이미지, 텍스트의 각 단어→컴퓨터 작업이 더 어려움

→ 딥러닝의 발적으로 인해 비구조적인 데이터를 해석하는 부분에서 크게 발

▼ 왜 딥러닝이 뜨고 있을까?

최근 데이터의 양(디지털 기기의 발달로 인함.)이 방대해 짐.

간단한 신경망은 데이터의 양이 증가해도 성능 향상의 한계가 있었으나 매우 큰 신경망을 훈련시키면 성능은 한계 없이 증가하게 됨.



높은 성능을 발휘하기 위해 필요한 것.

- 많은 양의 데이터
- 충분히 큰 신경망(많은 양의 은닉 유닛, 많은 연결과 많은 파라미터)

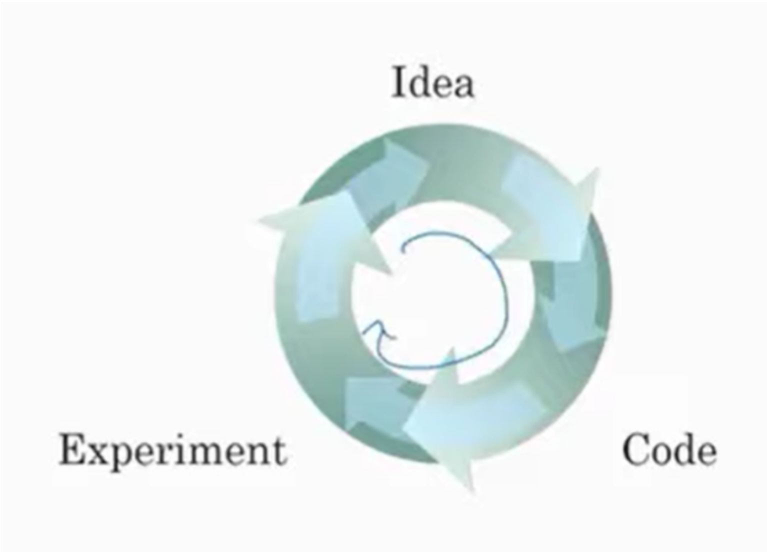
훈련할 데이터의 양이 많지 않으면 성능이 잘 비교가 안 됨. 특성을 다루는 실력이나 알고리즘의 작은 부분이 성능을 결정.

초창기 딥러닝의 문제

→데이터와 계산의 규모

→Data, **Computation(GPU, CPU등의 하드웨어 분야의 발전), Algorithms(exReLu의 도입으로 경사 하강법 속도 향상)**의 발전으로 크게 개선됨.

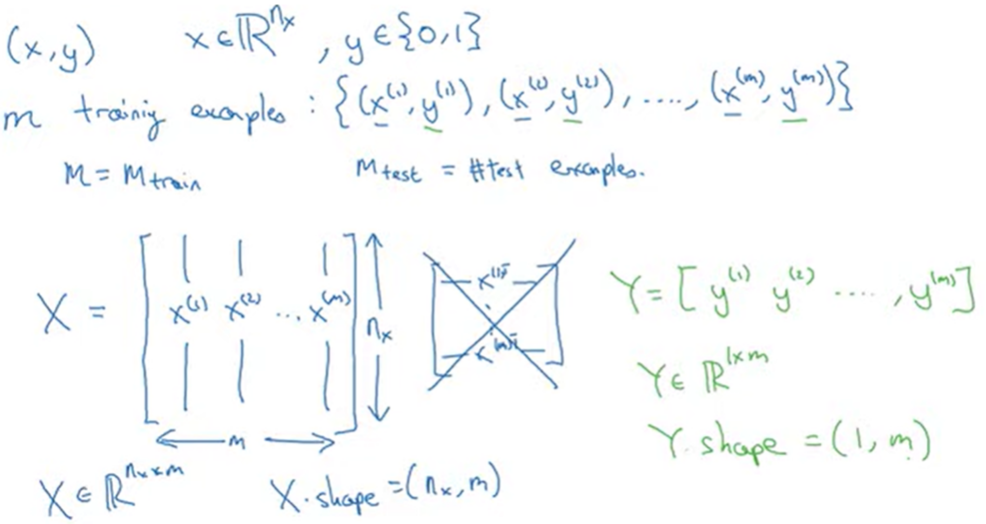
→ 계산 속도가 빨라지는 것이 중요함. 아이디어를 코드로 만들고 실험하는 과정의 시간을 단축시킬 수 있음.



▼ 이진 분류

ex 사진이 주어질 때, 고양이 여부

차원을 표기하는 방법

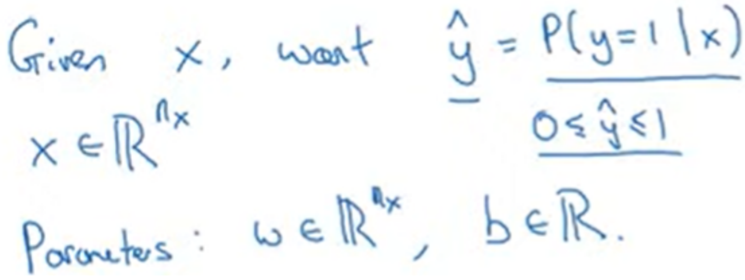


x=n_x(차원의 수)*m(데이터의 수)

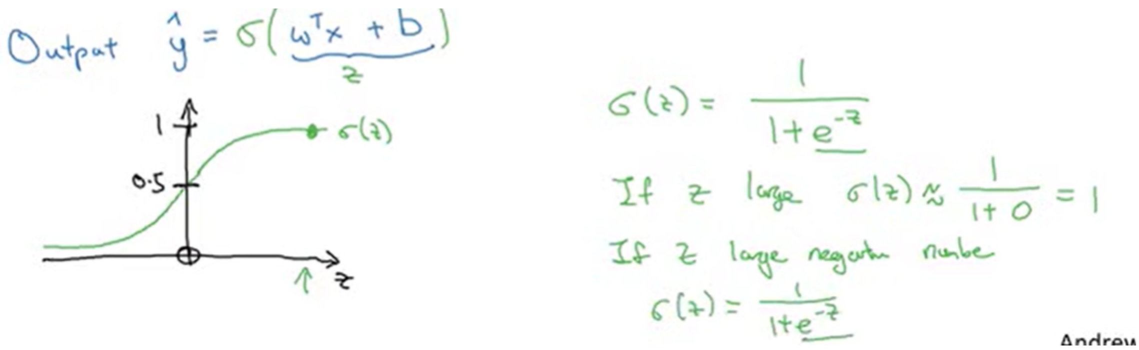
y=1*m

▼ 로지스틱 회귀

이진 분류에서 y의 예측값은 입력 특성 x가 주어졌을 때 y가 1일 **확률(항상 0과 1사이의 값이어야 함)**

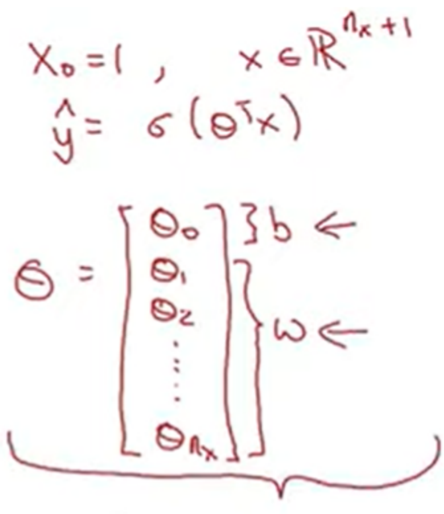


값의 범위를 0과1사이로 만들기 위해서 시그모이드 함수를 적용



→결과를 잘 예측하도록 w, b를 학습함.

아래의 표기로 w와b를 합쳐서 표현하기도 함.



▼ 로지스틱 회귀의 비용함수

로지스틱 회귀의 목적

$$\rightarrow \hat{y}^{(i)} = \sigma(w^T x^{(i)} + b), \text{ where } \sigma(z^{(i)}) = \frac{1}{1+e^{-z^{(i)}}} \qquad z^{(i)} = w^T x$$

$$\text{Given } \{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}, \text{ want } \hat{y}^{(i)} \approx y^{(i)}.$$

매개 변수들을 학습하기 위해 풀어야 하는 최적화함수가 볼록하지 않기 때문에 아래의 손실함수는 사용하지 않음. 여러 개의 지역 최적값을 가지고 있게 되어 문제 생기기에.

$$\text{Given } \{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}, \text{ want } \hat{y}^{(i)} \approx y^{(i)}.$$

$$\text{Loss (error) function: } \mathcal{L}(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2$$

Q. 최적화함수가 볼록하지 않다는 게 무슨 의미일까.

로지스틱 회귀에서는 아래와 같은 손실 함수를 사용한다.

$$\mathcal{L}(\hat{y}, y) = -\left(y \log \hat{y} + (1-y) \log(1-\hat{y})\right)$$

왜 이런 손실함수를 사용하는 것일까.

$$\text{If } y=1: \mathcal{L}(\hat{y}, y) = -\log \hat{y} \leftarrow \text{Want } \log \hat{y} \text{ large, want } \hat{y} \text{ large}$$

$$\text{If } y=0: \mathcal{L}(\hat{y}, y) = -\log(1-\hat{y}) \leftarrow \text{Want } \log 1-\hat{y} \text{ large} \ldots \text{ want } \hat{y} \text{ small}$$

- Loss(error) function(손실함수): 훈련 샘플 **하나**에 관하여 정의돼서 그 하나가 얼마나 잘 예측 되었는지 측정해줌.

- Cost function(비용함수): 훈련 세트 **전체**에 대해 얼마나 잘 추측되었는지 측정해 주는 함수.

$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m}\mathcal{L}(\hat{y}^{(i)}, y^{(i)}) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\hat{y}^{(i)} + (1-y^{(i)})\log(1-\hat{y}^{(i)})\right]$$

▼ 경사하강법

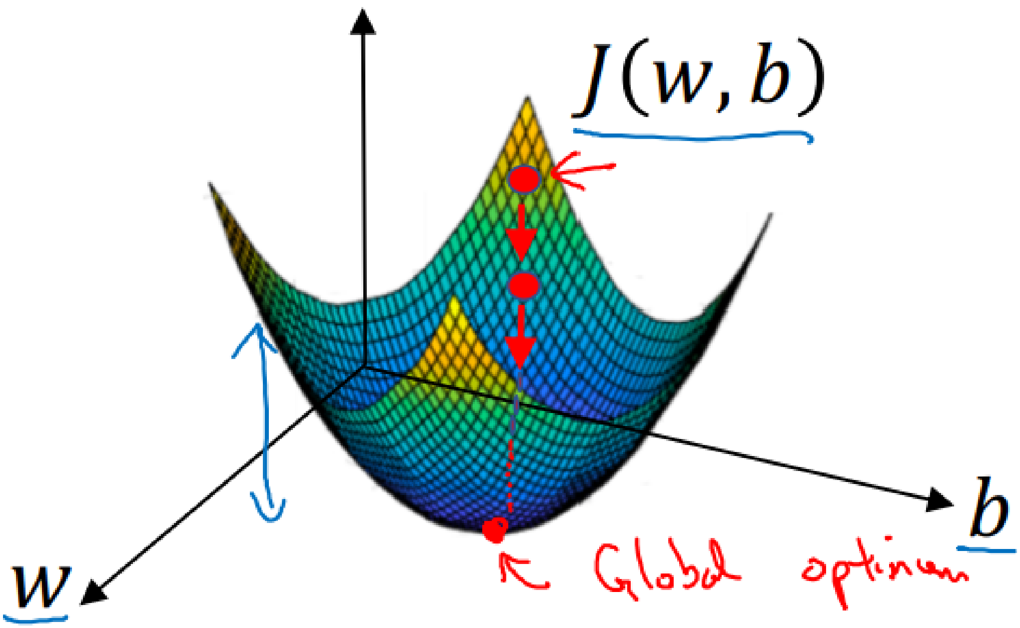
손실함수, 비용함수: 매개변수 w, b가 훈련세트를 얼마나 잘 예측하는지 측정

경사하강법 알고리즘: 매개변수 w, b를 훈련 세트에 학습시키는 방법

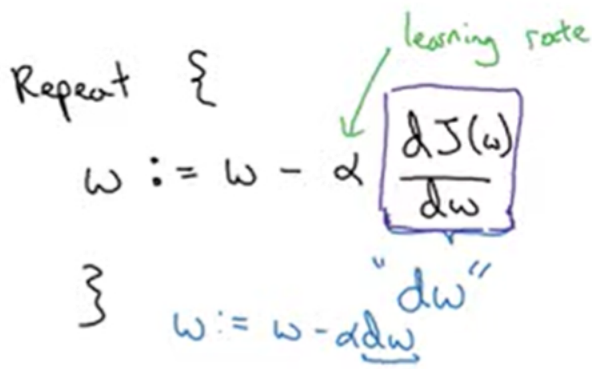
$$\text{Recap: } \hat{y} = \sigma(w^T x + b), \ \sigma(z) = \frac{1}{1+e^{-z}}$$

$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m}\mathcal{L}(\boxed{\hat{y}^{(i)}}, \boxed{y^{(i)}}) = -\frac{1}{m}\sum_{i=1}^{m} y^{(i)}\log\hat{y}^{(i)} + (1-y^{(i)})\log(1-\hat{y}^{(i)})$$

비용함수 J(w,b)가 볼록하다는 사실이 로지스틱 회귀에 위의 비용함수 J를 사용한 이유



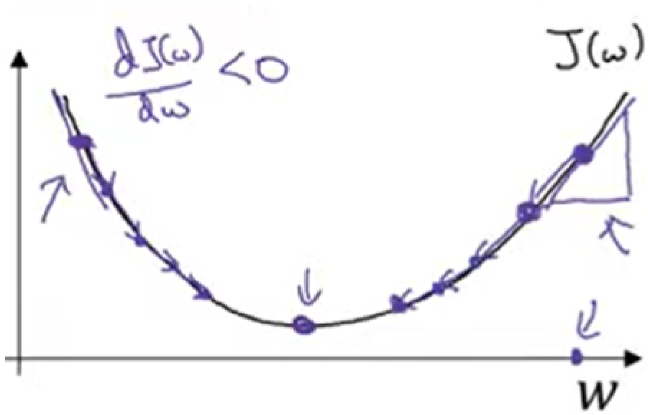
Gradient Descent



도함수는 함수의 기울기

만약 dw >0 이면, 파라미터 w 는 기존의 w 값 보다 작은 방향으로 업데이트

만약 dw <0 이면, 파라미터 w 는 기본의 w 값 보다 큰 방향으로 업데이트



- $w : w - \alpha \dfrac{dJ(w,b)}{dw}$
- $b : b - \alpha \dfrac{dJ(w,b)}{db}$

- α : 학습률이라고 하며, 얼만큼의 스텝으로 나아갈 것인지 정합니다.

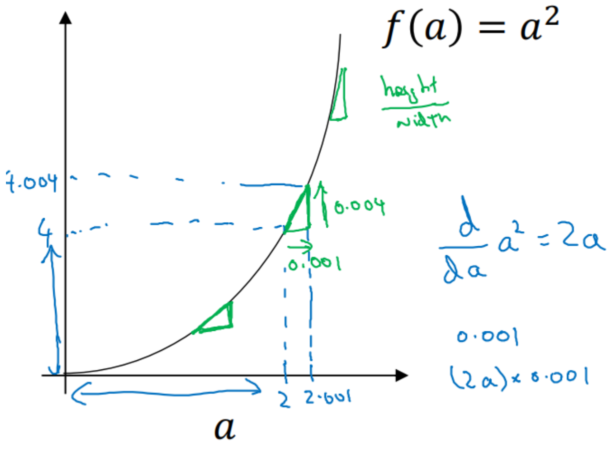
참고) 편미분 기호: **여러 변수** 중 하나에 대한 함수의 기울기를 구할 때

참고) 코드 구현 시 관습적으로 w의 변화를 나타내는 값→dw. b의 변화량을 나타내는 값→db

▼ 미분

도함수(어떤 함수의 기울기): 변수 a를 조금만 변화했을 때, 함수 f(a) 가 얼만큼 변하는 지는 측정하는 것

▼ 더 많은 미분 예제



$$\frac{d}{da} a^2 = 2a$$

0.001

$(2a) \times 0.001$

$a = 2$ $\quad f(a) = 4$

$a = 2.001$ $\quad f(a) \approx 4.004$ (4.004001)

slope (derivative) of f(a) at a=2 is 4.

$\boxed{\frac{d}{da} f(a) = 4}$ when $\boxed{a=2}$.

$a = 5$ $\quad f(a) = 25$

$a = 5.001$ $\quad f(a) \approx 25.010$

$\boxed{\frac{d}{da} f(a) = 10}$ when $\boxed{a=5}$

$$\frac{d}{da} f(a) = \frac{d}{da} a^2 = \boxed{2a}$$

→ 실제의 오차는 a근처로 무한소 가까이 민 경우, 사라지게 된다.

더 많은 예

$f(a) = a^2$ $\quad \frac{d}{da} f(a) = \underbrace{2a}_{4}$ $\quad a = 2$ $\quad f(a) = 4$

$\quad a = 2.001$ $\quad f(a) \approx 4.004$

$f(a) = a^3$ $\quad \frac{d}{da} f(a) = 3a^2$ $\quad 3 \times 2^2 = 12$ $\quad a = 2$ $\quad f(a) = 8$

$\quad a = 2.001$ $\quad f(a) \approx 8.012$

$f(a) = \log_e(a)$ $\quad \frac{d}{da} f(a) = \frac{1}{a}$ $\quad a = 2$ $\quad f(a) \approx 0.69315$

$\ln(a)$ $\quad a = 2.001$ $\quad f(a) \approx 0.69365$

$\frac{d}{da} f(a) = \boxed{\frac{1}{2}}$

0.0005 ⟷ 0.0005