

부스트코스 딥러닝 1단계: 신경망과 딥러닝

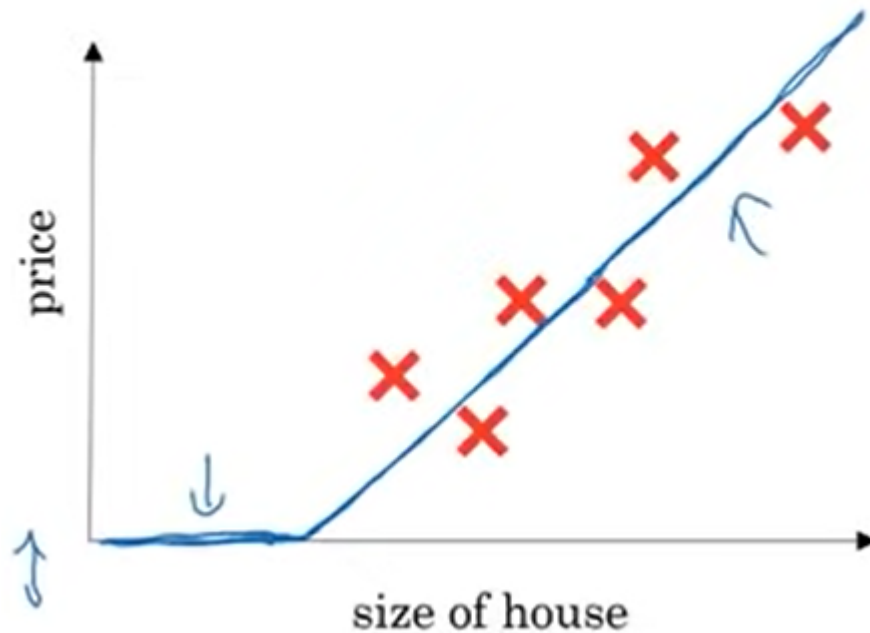
딥러닝 소개

신경망이란 무엇인가?

ex 집값 문제

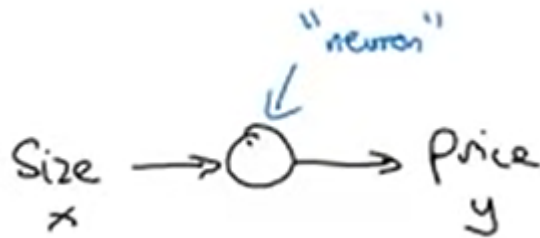
feature : price, size of house

선형 회귀로 상관관계를 예측할 수 있다.



이걸 신경망으로 나타낸다면?

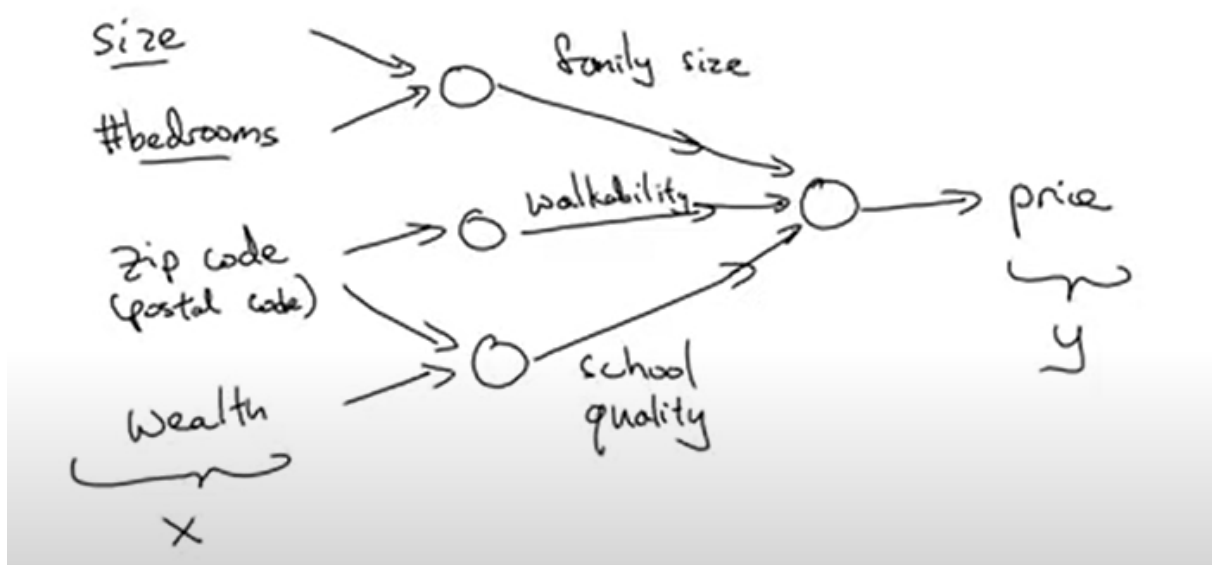
하나의 신경으로 나타낼 수 있다.



그런데, 위에 선형함수는 ReLU 의 모양을 하고 있다. ReLU란 Rectified Linear Unit.



침실 수, 가족의 크기, 평방미터 등 집값을 결정하는 요소가 많다면? 다중 신경망으로 나타낼 수 있다. 중앙에 있는 것들은 스스로 알아낸다.



지도학습

음성인식, 자산, 광고, 자율주행, 기계 번역, 부동산 분야에 적합하다.

부동산 어플리케이션, 광고 - 표준 신경망 구조

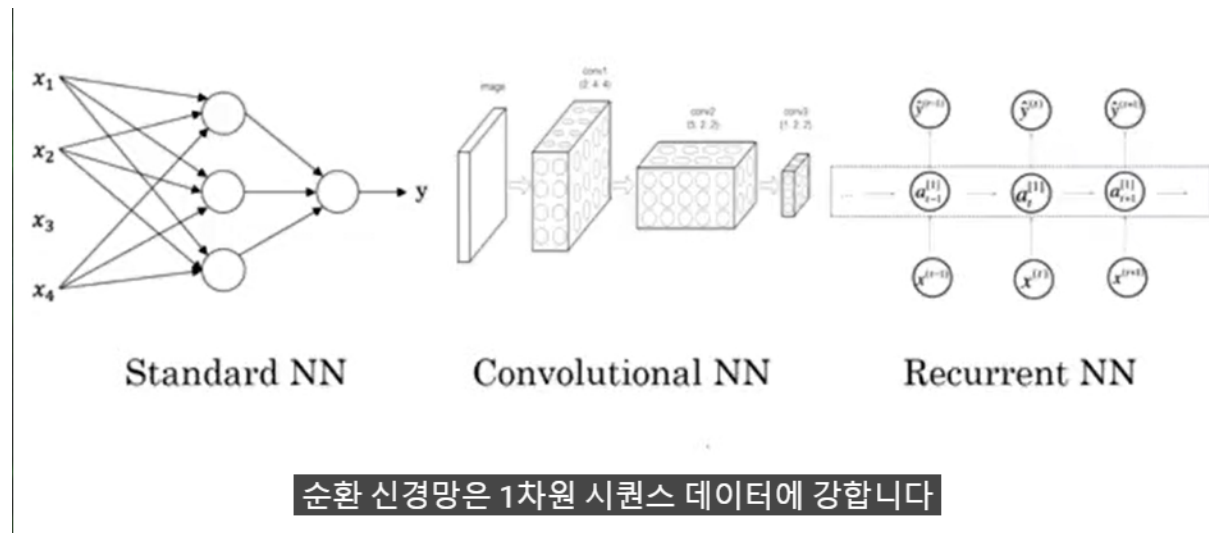
이미지 분야 - CNN (합성곱 신경망)

음성 인식, 언어 - 1차원 시계열 데이터로 나타나는 시계열 데이터: RNN(순환신경망)

자율주행 - CNN

레이더 정보 - 하이브리드 구조

CNN RNN 의 기준



구조적 데이터 VS 비구조적 데이터

구조적 데이터 - 데이터베이스 구조

Structured Data

Size	#bedrooms	...	Price (1000\$s)
2104	3		400
1600	3		330
2400	3		369
⋮	⋮		⋮
3000	4		540

User Age	Ad Id	...	Click
41	93242		1
80	93287		0
18	87312		1
⋮	⋮		⋮
27	71244		1

구조적 데이터는

비구조적 데이터 - 음성파일, 이미지, 텍스트

추출하기 어려운 형태의 데이터이다. 딥러닝 덕분에 컴퓨터가 비구조적 데이터를 인식할 수 있게 되었다.

Unstructured Data



Audio



Image

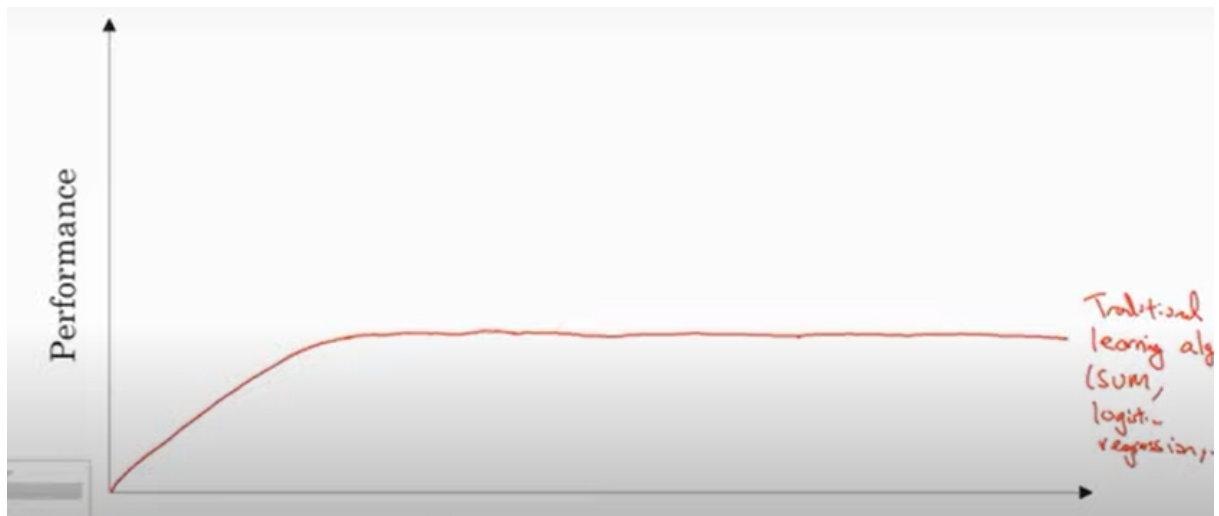
Four scores and seven
years ago...

이미지의 픽셀값이나

Text

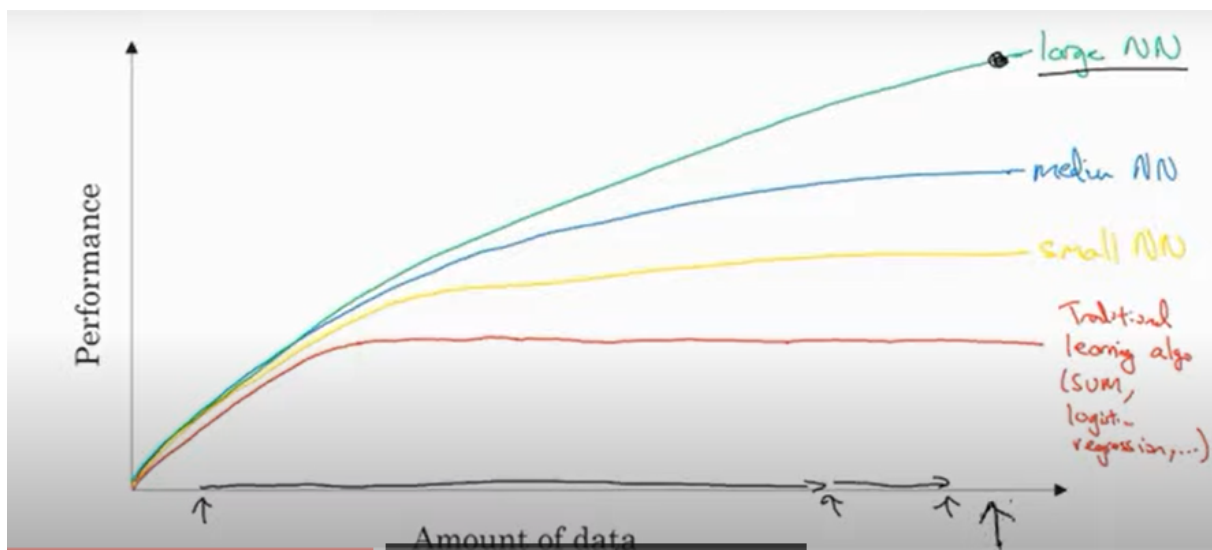
왜 이제서야 뜨는가?

딥러닝 연구는 예전부터 진행되고 있었다.



데이터 양과 성능 간의 관계 그래프

딥러닝의 성장동력



- 많은 양의 데이터를 이용하기 위한 충분히 큰 신경망(여기서 데이터는 라벨이 있는 데이터:m)
- 데이터의 크기

⇒ 단순히 규모를 키우는 것만으로도 성능을 높일 수 있다.

⇒ 데이터의 양이 적으면 알고리즘을 어떤걸 쓰느냐가 꽤 중요하지만, 데이터가 커질수록 규모가 중요해진다.

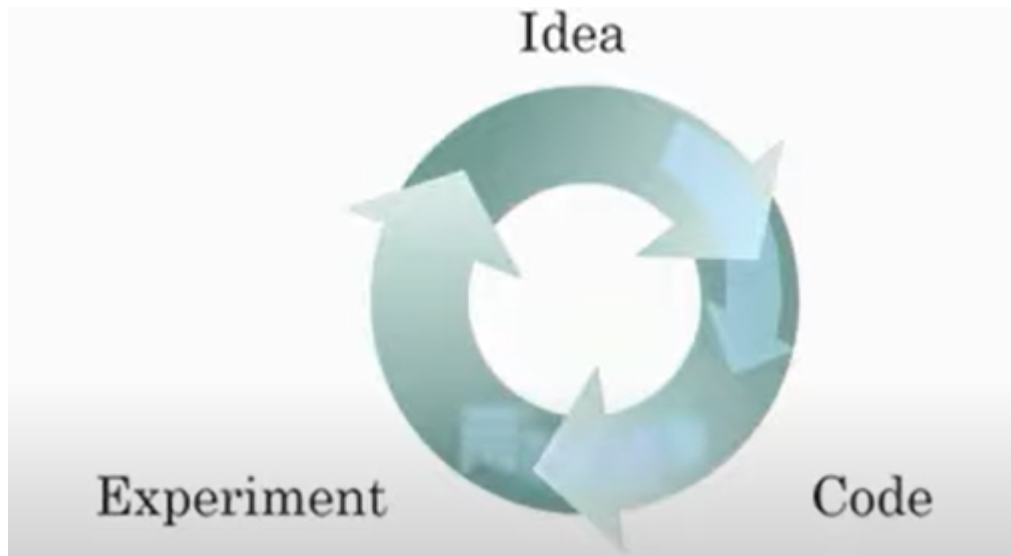
초창기 문제

- 데이터
- 계산능력
- 알고리즘

⇒ 시그모이드 함수에서 ReLU함수로 바꾼 것이 성능을 향상시키는데 혁신적이었다. 왜냐하면 경사가 0일 때 경사 하강법에서 급격히 느려지기 때문에 학습이 오래걸리기 때문이다.

빠른 계산이 중요한 이유

신경망을 학습시키는 과정이 반복적이기 때문이다. ⇒ 결국 생산성에 영향을 준다. 10분 vs 하루 vs 한달의 차이



신경망과 로지스틱회귀

키워드

for loop을 쓰지 않고 데이터를 처리하는 방법

정방향 패스 역방향 패스

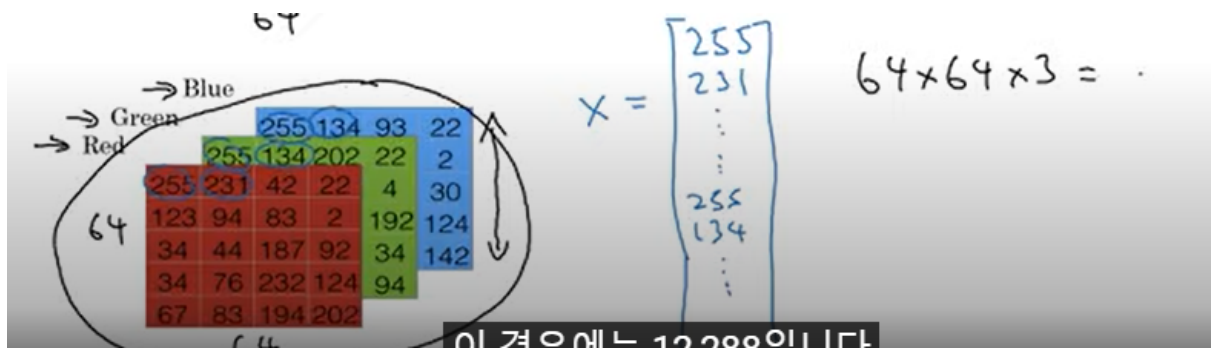
로지스틱 회귀(이진분류를 위함)

이진 분류

그렇다/아니다 두개로 분류하는 것이다.

예시) 고양이인지 아닌지 구분하는 문제

feature vector 에 담는다



$(x, y) \quad x \in \mathbb{R}^{n_x}, y \in \{0, 1\}$

m training examples: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$

$M = M_{\text{train}} \quad M_{\text{test}} = \# \text{test examples.}$

$X = \begin{bmatrix} | & | & & | \\ x^{(1)} & x^{(2)} & \dots & x^{(m)} \\ | & | & & | \end{bmatrix}$

$X \in \mathbb{R}^{n_x \times m} \quad X.\text{shape} = (n_x, m)$

$Y = [y^{(1)} \ y^{(2)} \ \dots \ y^{(m)}]$

$Y \in \mathbb{R}^{1 \times m}$

$Y.\text{shape} = (1, m)$

행렬과 벡터로 나타내면 위와 같다.

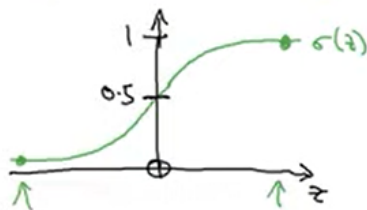
로지스틱 회귀

이진분류 문제에서 많이 쓰인다

Given x , want $\hat{y} = \frac{P(y=1|x)}{0 \leq \hat{y} \leq 1}$
 $x \in \mathbb{R}^{n_x}$

Parameters: $\underline{w} \in \mathbb{R}^{n_x}$, $\underline{b} \in \mathbb{R}$.

Output $\hat{y} = \sigma(\underline{w}^T x + b)$



$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

If z large $\sigma(z) \approx \frac{1}{1+0} = 1$

If z large negative number

$$\sigma(z) = \frac{1}{1 + e^{-z}} \approx \frac{1}{1 + \text{Big num}} \approx 0$$

표기법에 대해 하나 덧붙이자면

Andrew

시그모이드 함수를 쓰는 이유: $w^T x + b$ 는 선형식이기 때문에 1보다 아주 큰 수가 나올텐데 결과는 0과 1사이에 분포 해야한다. 따라서 0과 1사이 값으로 만들어주는 시그모이드 함수가 필요하다.

$$x_0 = 1, \quad x \in \mathbb{R}^{n_x + 1}$$

$$\hat{y} = \sigma(\theta^T x)$$

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_{n_x} \end{bmatrix} \left\{ \begin{array}{l} b \leftarrow \\ w \leftarrow \end{array} \right.$$

weight 와 bias 를 한 벡터에 넣으면 위와 같다.

로지스틱 회귀의 비용함수

매개변수를 학습하려면 비용함수에 대해 알아봐야한다.

Logistic Regression cost function

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

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

Loss (error) function:

$$\ell(\hat{y}, y) = \frac{1}{2} (\hat{y} - y)^2$$

$$\ell(\hat{y}, y) = - (y \log \hat{y} + (1-y) \log (1-\hat{y})) \leftarrow$$

If $y=1$: $\ell(\hat{y}, y) = -\log \hat{y} \leftarrow$ Want $\log \hat{y}$ large, want \hat{y} large.

If $y=0$: $\ell(\hat{y}, y) = -\log (1-\hat{y}) \leftarrow$ Want $\log (1-\hat{y})$ large ... Want \hat{y} small

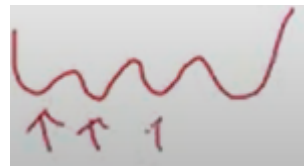
Cost function: $J(w, b) = \frac{1}{2m} \sum_{i=1}^m \ell(\hat{y}^{(i)}, y^{(i)}) = -\frac{1}{2m} \sum_{i=1}^m [y^{(i)} \log \hat{y}^{(i)} + (1-y^{(i)}) \log (1-\hat{y}^{(i)})]$

손실 함수 J를 최소화해주는 매개 변수들 w와 b를 찾는 것입니다

convex(아래로 볼록이 나오는 제곱 오차를 로지스틱 회귀에서 쓰지 않는 이유) : 매개변수들을 학습하기 위해 풀어야 할 최적화 함수가 볼록하지 않기 때문이다. 그러므로 여러개의 지역 최적값을 가지고 있게 되어 문제가 생긴다. 이런 경우 경사하강법으로 최적의 값을 찾을 수 없다.



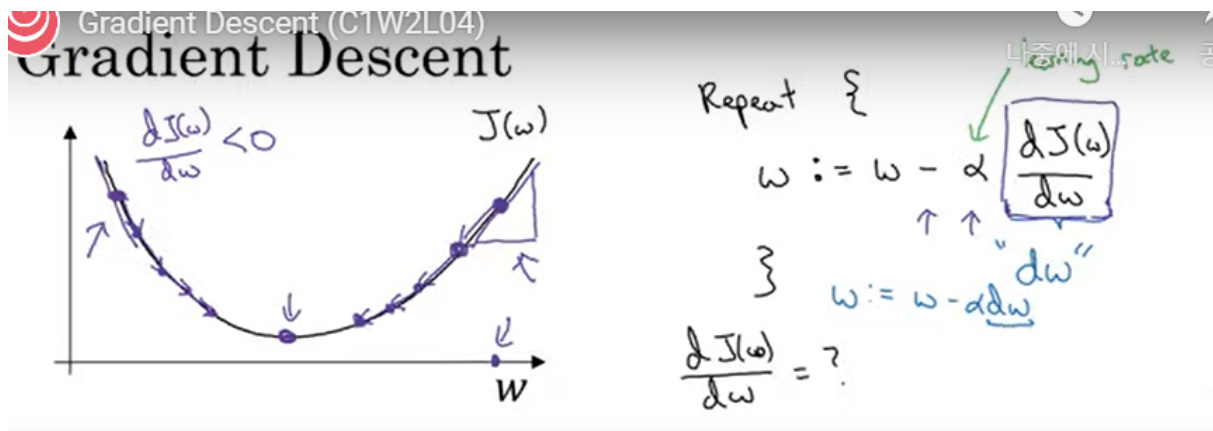
convex



non-convex

경사 하강법

매개변수 w 와 b를 훈련세트에 학습 시키는 법을 찾는다.



현재 매개변수 위치에서 미분계수를 찾아 비용함수 J에서 가장 가파르게 내려가는 방향을 알 수 있다.

$$J(w, b) \quad w := w - \alpha \frac{\partial J(w, b)}{\partial w}$$

$$b := b - \alpha \frac{\partial J(w, b)}{\partial b}$$

w, b 각각을 구한다.

$$w := w - \alpha \frac{\partial J(w, b)}{\partial w}$$

$$b := b - \alpha \frac{\partial J(w, b)}{\partial b}$$

이것은 단순히 J(w,b)의 미분계수를 의미합니다

J(w,b) 가 w방향으로 얼마나 기울었는지를 나타낸다.

② "partial derivatin"

변수가 두개면 위의 기호를 쓴다.

현재 w, b 두개의 변수를 구해야하니 편미분 기호를 쓴다.

$$\frac{\partial J(w, b)}{\partial w}$$

$$\frac{\partial J(w, b)}{\partial b}$$

합나다

② "partial derivatin"

① J

dw

db

Andre

미분(derivatives)

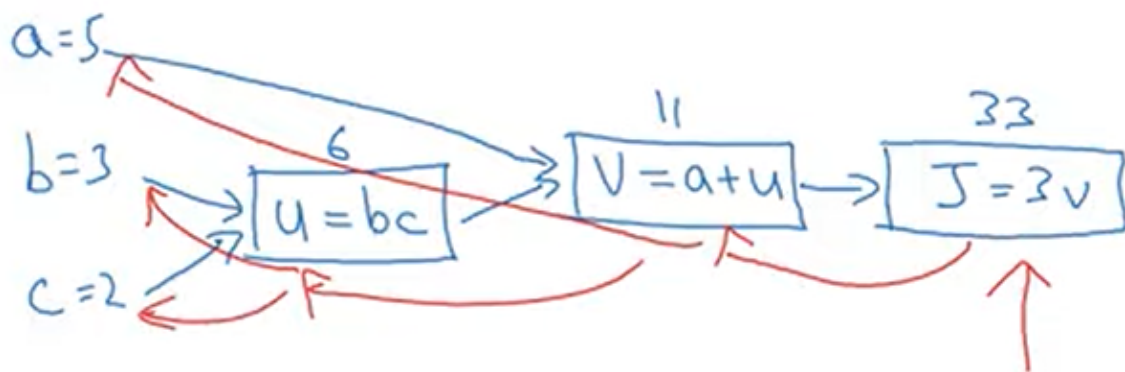
변수와 함수의 변화 비율

$$\frac{df(a)}{da} = 3 = \frac{d}{da} f(a)$$

a 가 1만큼 변할 때, f(a) 는 3만큼 변한다.

계산 그래프

함성함수를 변수에서 출발하여 그래프로 만드는 법

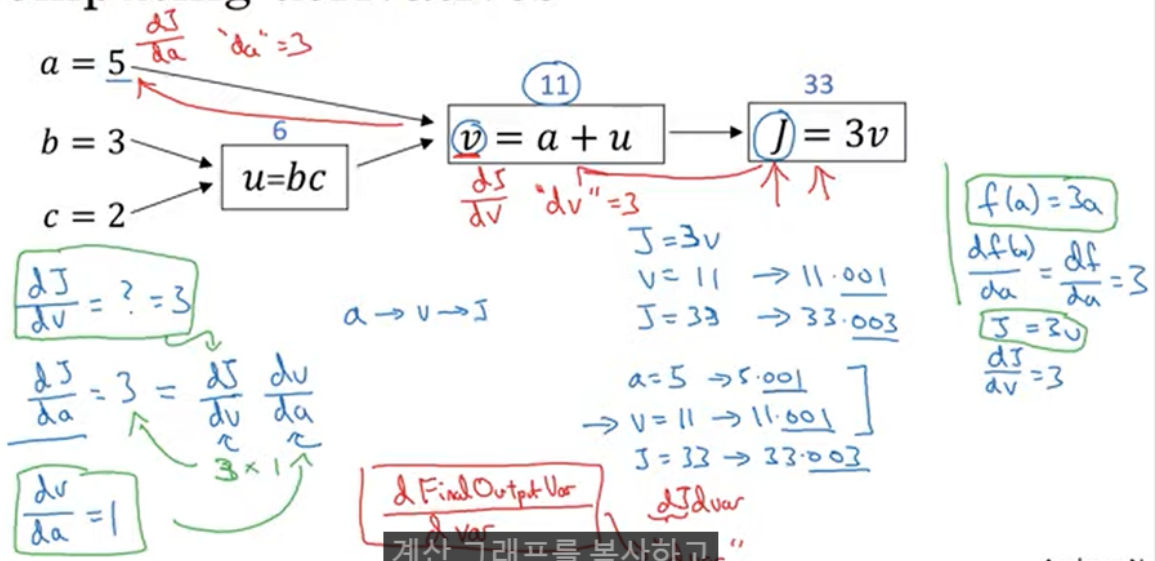


파란색 : 정방향 계산

빨간색 : 역방향, 도함수를 계산

계산 그래프

Computing derivatives



$$dJ/da == da$$

$$dJ/dv == dv$$

$$dJ/du == du$$

$$dJ/db == db = dJ/dv * dv/du * du/db = 3 * 1 * c(2) = 3c(6)$$

$$\frac{d \text{ Final output var}}{d \text{ var}} = d \text{ var}$$

표기법

로지스틱 회귀의 경사 하강법

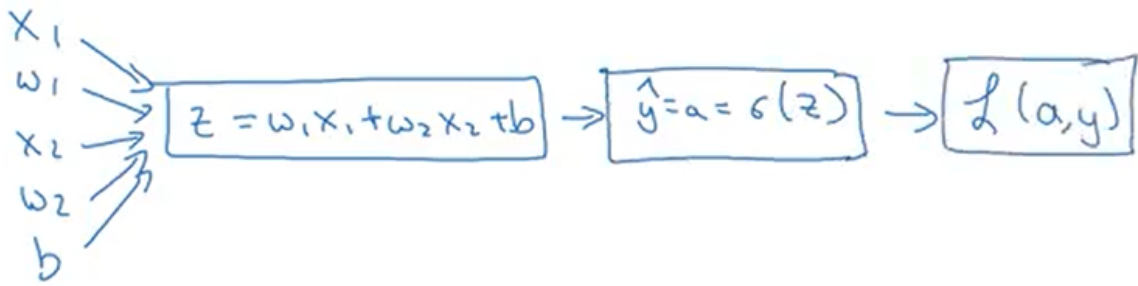
Logistic regression recap

$$\rightarrow z = w^T x + b$$

$$\rightarrow \hat{y} = a = \sigma(z)$$

$$\mathcal{L}(a, y) = -(y \log(a) + (1 - y) \log(1 - a))$$

recap



특성이 x_1, x_2 라고 가정 $\Rightarrow w_1, w_2, b$ 필요

m개 샘플의 경사 하강법

한개의 샘플이 아닌 m개 샘플을 경사하강법을 쓰는 수식

$$\begin{aligned}
 &J=0; \quad dw_1=0; \quad dw_2=0; \quad db=0 \\
 &\text{For } i=1 \text{ to } m \\
 &\quad z^{(i)} = w^T x^{(i)} + b \\
 &\quad a^{(i)} = \sigma(z^{(i)}) \\
 &\quad J += -[y^{(i)} \log a^{(i)} + (1-y^{(i)}) \log(1-a^{(i)})] \\
 &\quad dz^{(i)} = a^{(i)} - y^{(i)} \\
 &\quad dw_1 += x_1^{(i)} dz^{(i)} \\
 &\quad dw_2 += x_2^{(i)} dz^{(i)} \\
 &\quad db += dz^{(i)} \\
 &\quad \updownarrow n=2 \\
 &J /= m \\
 &dw_1 /= m; \quad dw_2 /= m; \quad db /= m.
 \end{aligned}$$

