<딥러닝 소개>

deeplearning.ai

Introduction to Deep Learning

Welcome

- AI is the new Electricity
- Electricity had once transformed countless industries: transportation, manufacturing, healthcare, communications, and more
- AI will now bring about an equally big transformation.

Andrew Ng

What you'll learn

Courses in this sequence (Specialization):

1. Neural Networks and Deep Learning
2. Improving Deep Neural Networks: Hyperparameter tuning, Regularization and Optimization
3. Structuring your Machine Learning project
4. Convolutional Neural Networks
5. Natural Language Processing: Building sequence models

Andrew Ng

Deep Learning 을 포함한 새로운 Neural Network 를 어떻게 설계하는지, 어떻게 데이터를 가지고 학습 하는지 배워서 최종적으로 고양이 인식기를 만들 것입니다.

NN 작동원리 및 하이퍼 파라미터 조율, bias와 variance는 어떻게 찾는지, 최적화 알고리즘은 무엇이 있는지 배울 것입니다.

ML 프로젝트를 어떻게 설계해야하는지 배울 것입니다.

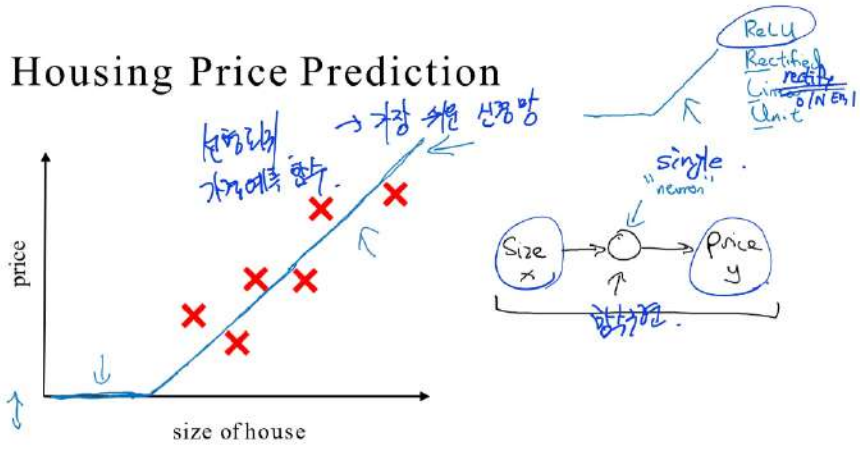
이미지에 적용되는 CNN 모델 구축 방법에 대해서 배웁니다.

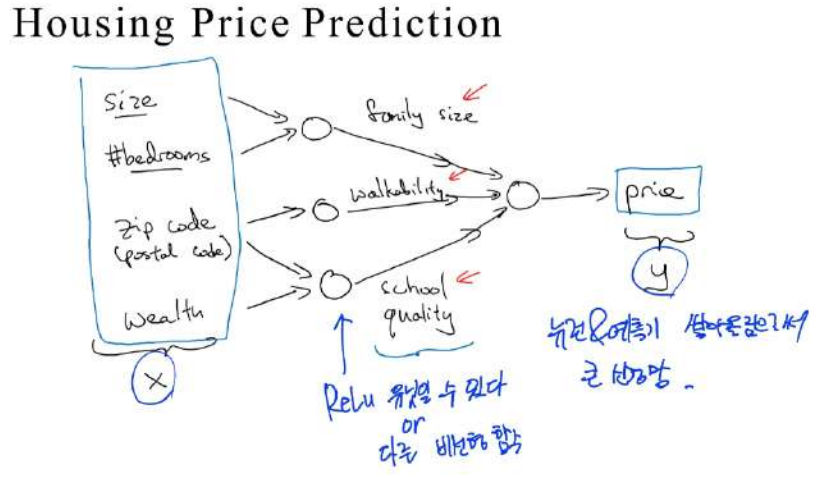
2.

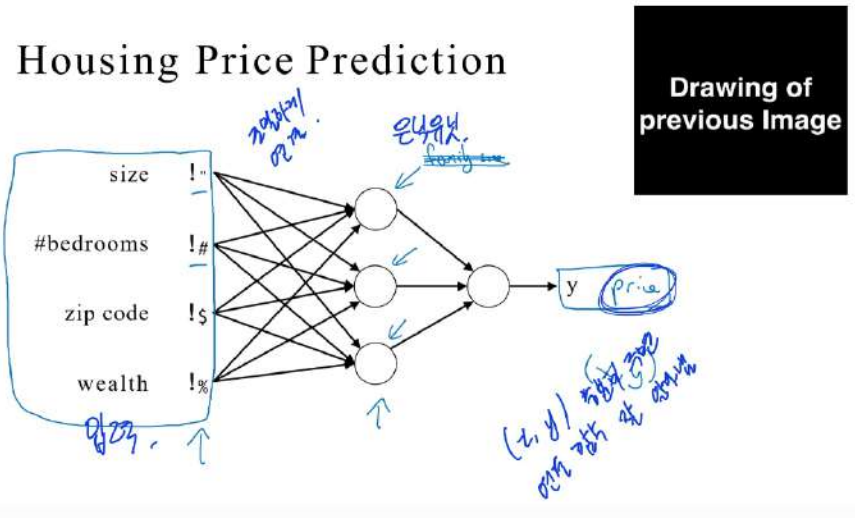
deeplearning.ai

Introduction to Deep Learning

What is a Neural Network?







Drawing of previous Image

신경망란 입력(x)와 출력(y)를 매칭해 주는 함수를 찾는 과정입니다.

충분한 데이터가 주어지면 더 잘 알아낼 수 있습니다.

해당 뉴런에 관계없는 입력값이라도 입력으로 넣어주어야 합니다. 관계 여부는 신경망이 학습하면서 알아서 조절해 줍니다.

deeplearning.ai

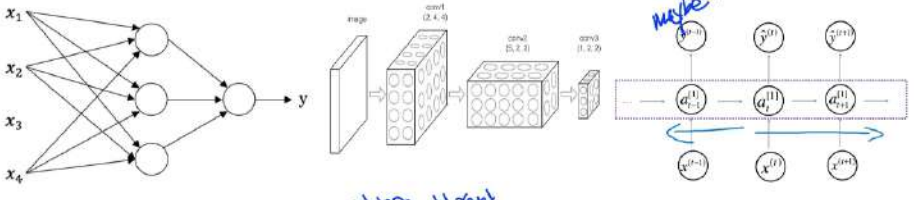
Introduction to Deep Learning

Supervised Learning with Neural Networks

Supervised Learning

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

Neural Network examples



Standard NN | Convolutional NN | Recurrent NN

Supervised Learning

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

Unstructured Data

Audio

Image

Four scores and seven years ago...

Text

- 머신러닝의 방법은 지도 학습, 비지도 학습 등 여러가지 종류가 있습니다.

- 지도 학습이란 정답이 주어져있는 데이터를 사용하여 컴퓨터를 학습시키는 방법을 뜻합니다.

- 앞 강의에서 배운 신경망을 이용해 지도 학습을 구현할 수 있습니다.

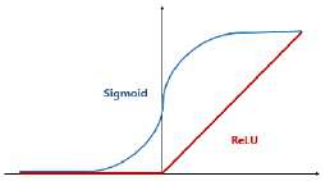
- 분야에 따라 적용되는 신경망이 다릅니다.

ex) 이미지 분류를 위해 CNN 사용, 음성을 텍스트로 변환시키기 위해 RNN 사용

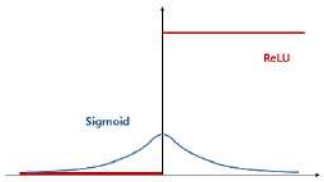
- 구조적 및 비구조적 데이터를 신경망을 사용하여 예측할 수 있습니다.

- 구조적 데이터: 데이터베이스로 표현된 데이터를 말합니다. 정보의 특성이 잘 정의되어 있습니다.

- 비구조적 데이터: 이미지, 오디오와 같이 특징적인 값을 추출하기 어려운 형태의 데이터입니다. 딥러닝 덕분에 컴퓨터가 비구조적 데이터를 인식할 수 있게 되었습니다.



(1) sigmoid와 ReLU 함수



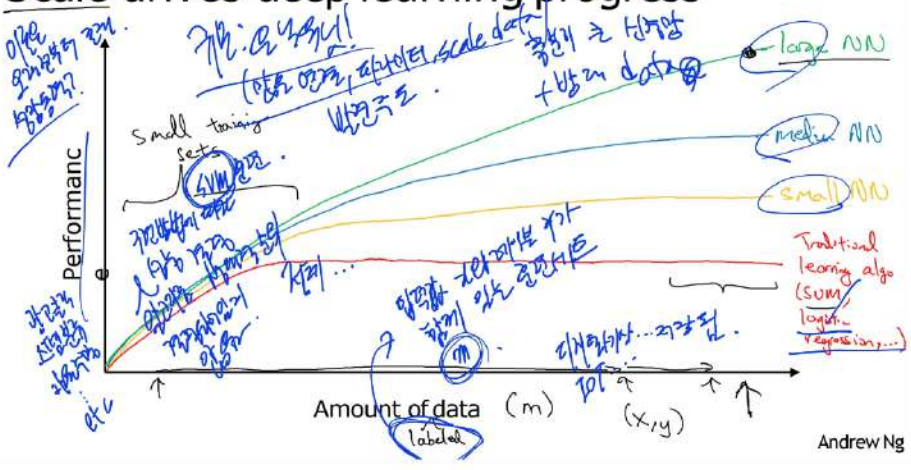
(2) sigmoid와 ReLU 함수의 미분

deeplearning.ai

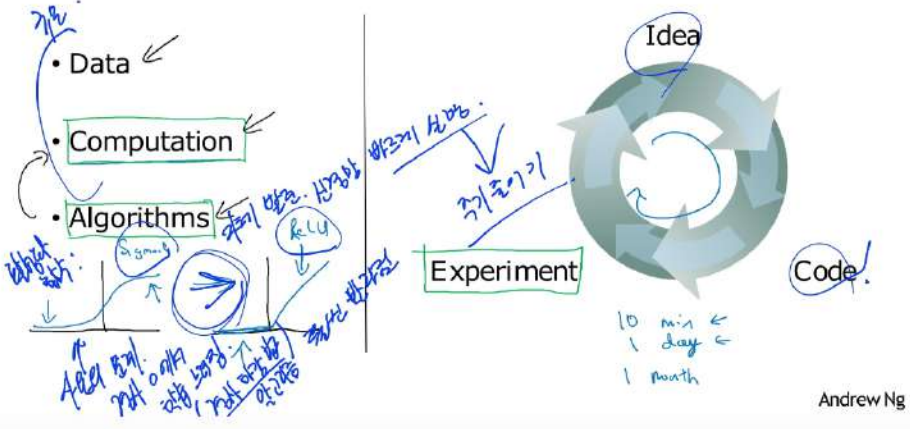
Introduction to Neural Networks

Why is Deep Learning taking off?

Scale drives deep learning progress



Scale drives deep learning progress



- 깊은 모델 일수록 더 많은 데이터가 필요하며, 이는 곧 좋은 퍼포먼스로 나타납니다.
- 최근 딥러닝이 강력한 도구로 부상한 이유는 아래의 3가지 요인들로 인해 딥러닝 성능이 향상되었기 때문입니다.
- 데이터 양 증가, 컴퓨터 성능 향상, 알고리즘의 개선

ex) Sigmoid 함수가 아닌 ReLU 함수를 사용함으로 Gradient 소멸 문제 해결

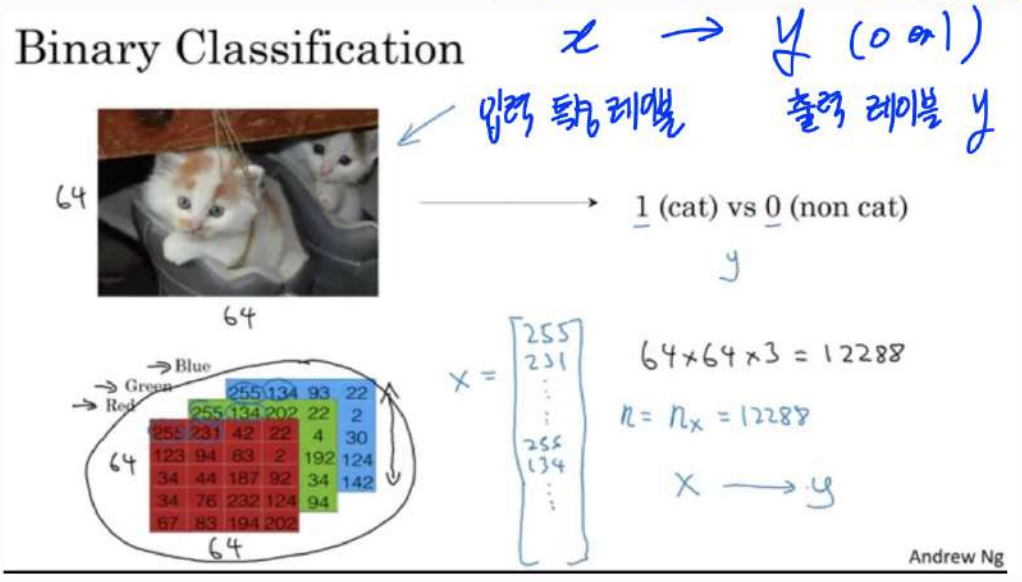
또한, 이전과 달리 빠른 실험 결과를 얻을 수 있어서, 아이디어(Idea) 생산 > (Code) 구현 > 실험(Experiment)결과의 시간이 단축돼서, 더 많은 아이디어를 실험 할 수 있게 됐습니다. 이는 오늘날 딥러닝 알고리즘 분야에서 경이로운 혁신으로 이어졌습니다.

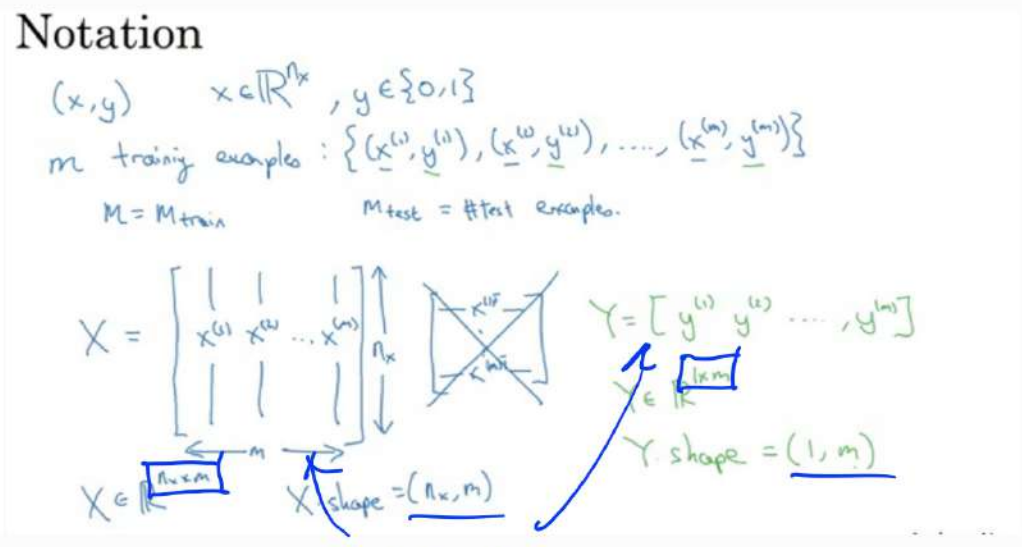
<신경망과 로지스틱 회귀>

m개 훈련세트 for loop 이용? … 정방향 전파/역 전파로 구성

로지스틱 회귀 : 이진 분류를 위한 알고리즘

$x \rightarrow y$ (0 or 1) / n_x 차원.





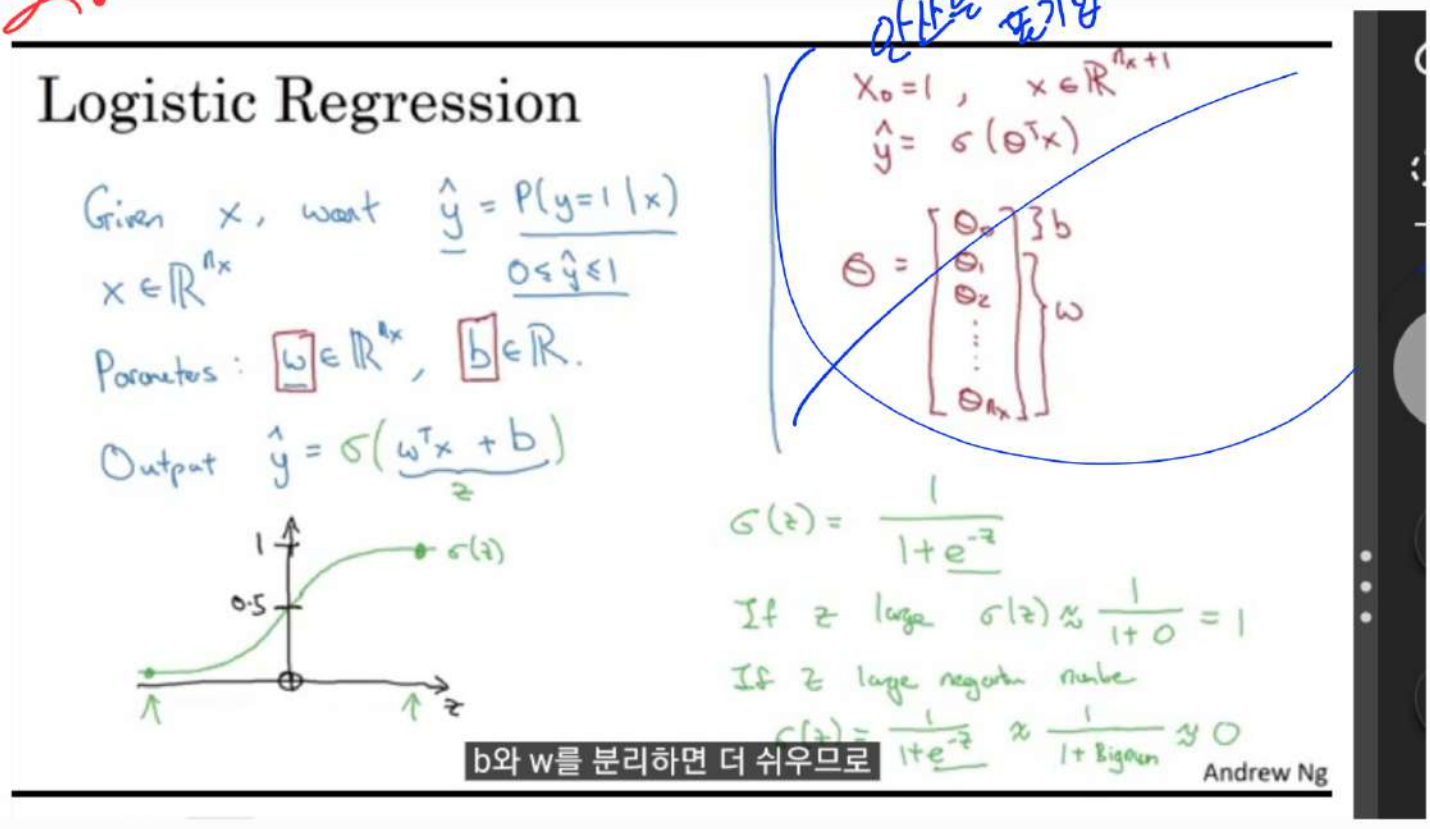
- 신경망에서 학습하는 방법은 정방향 전파와 역전파가 있습니다. 향후 몇개의 수업에서 로지스틱 분류로 신경망이 왜 전방향 전파와 역전파로 구성되어 있는지 직관을 얻을 것입니다.

- 이진 분류란 그렇다 / 아니다 2개로 분류하는 것입니다. 이때 결과가 '그렇다' 이면 1로 표현하고 '아니다'이면 0으로 표현합니다.

ex) 고양이다 / 고양이가 아니다.

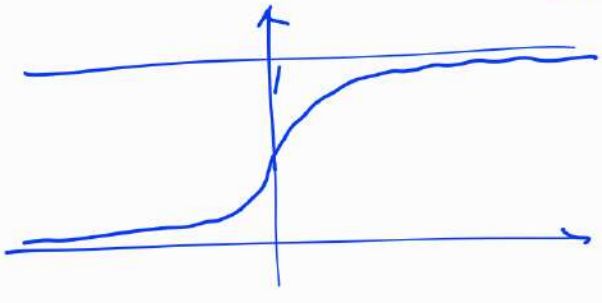
- 로지스틱 회귀란 이진 분류를 하기 위해 사용되는 알고리즘입니다.

2.



- 로지스틱 회귀란 답이 0 또는 1로 정해져있는 이진 분류 문제에 사용되는 알고리즘입니다.
- X (입력 특성), y (주어진 입력특성 X에 해당하는 실제 값) y^ (y의 예측값)을 의미합니다.
- 더 자세히 이진 분류를 위한 y^ 값은 y가 1일 확률을 의미하며 0 ≤y^≤ 1 사이의 값을 가져야 합니다.
- 선형 회귀시 $\hat{y} = W^T X+b$ 를 통해 계산하지만, 해당 값은 0과 1 범위를 벗어날 수 있습니다. 따라서 시그모이드 함수를 통해 0과 1사이의 값으로 변환해줍니다.
- 따라서 로지스틱 회귀를 위한 $\hat{y}=\sigma(W^T X+b)$ 로 구하게 됩니다.

참고) 시그모이드 함수 sigma(z)=~~1+e-z~~ $\frac{1}{1+e^{-z}}$



3.

Given x want $\hat{y} = P(y=1|x)$

파라미터 w, b 학습하기 위해 비용함수를 정의해야 한다.

Logistic Regression cost function

$\rightarrow \hat{y}^{(i)} = \sigma(w^T x^{(i)} + b)$, where $\sigma(z^{(i)}) = \frac{1}{1+e^{-z^{(i)}}}$ — $x^{(i)}$: i번째 훈련샘플. $\quad z^{(i)} = w^T x^{(i)} + b$

예측값 시그모이드.

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

$x^{(i)}$, $y^{(i)}$, $z^{(i)}$: i-th example.

예측값 참값 오차 얼마나 클지. → 로지스틱에서는 경사하강법 사용X

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

알고리즘 얼마나 잘 작동하는지

최적화 함수가 볼록하지 않다! (여러 개의 최적값) 지역. → 경사하강법이 전역 최적값은 찾지 못함.

Single training set.

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

오차 값 줄이자!

① If $y=1$: $\mathcal{L}(\hat{y}, y) = -\log \hat{y}$ ← want $\log \hat{y}$ large, want $\hat{y}$ large (1에 수렴하길 원한다)

② If $y=0$: $\mathcal{L}(\hat{y}, y) = -\log(1-\hat{y})$ ← want $\log(1-\hat{y})$ large .... want $\hat{y}$ small (0에 가깝).

매개변수의 비용 / 최소화 하는

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

훈련세트 전체에 대해 얼마나 잘 추측되었는지 측정.

Andrew Ng

최적화 위해 볼록한 함수 다시 정의!

새로운 손실함수 → 훈련 샘플 하나에 관해 정의되어서 얼마나 잘 예측되었는지 측정

①,② 볼록 함수 된다.. 비교하고 정리. z 하나가

parameters $w \in \mathbb{R}^{n_x}$, $b \in \mathbb{R}$

Output $\hat{y} = \sigma(w^T x + b)$ 0~1

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

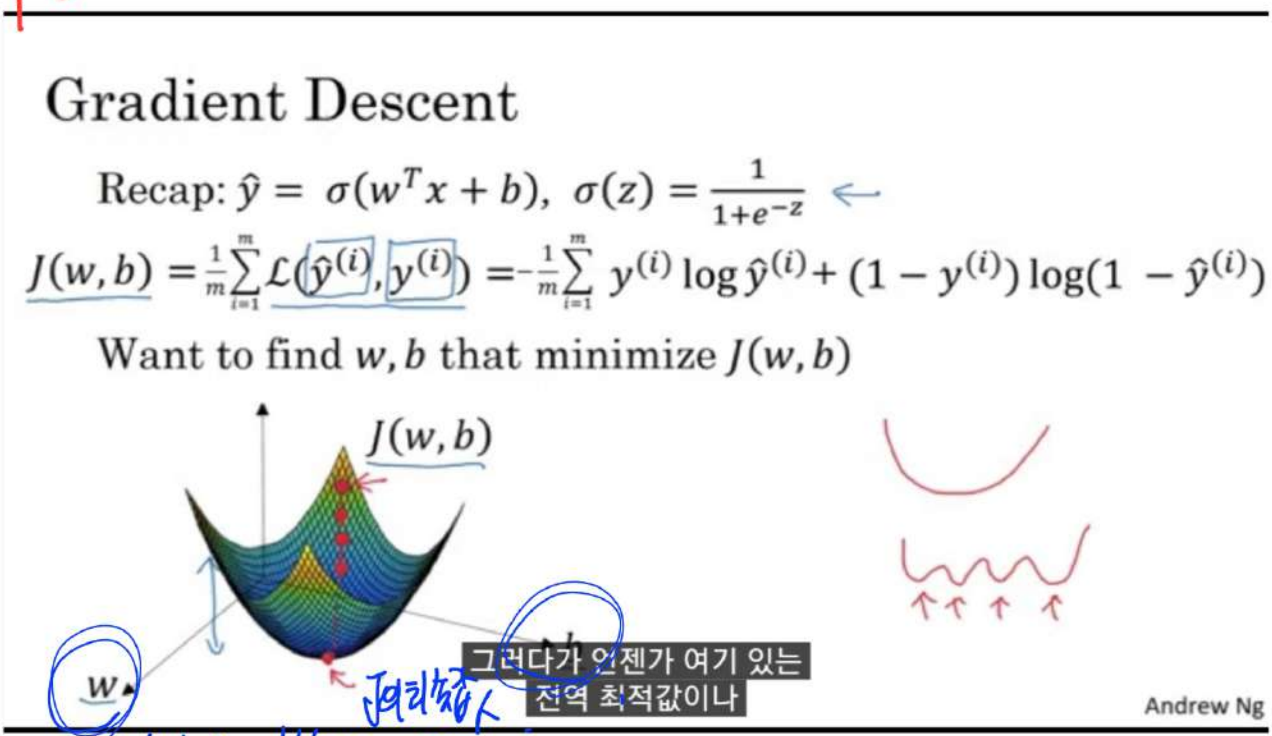
if z large $\sigma(z) \approx 1$

small ≈ 0

y가 1일 확률 잘 계산하도록 파라미터 w, b 값 학습시켜야!

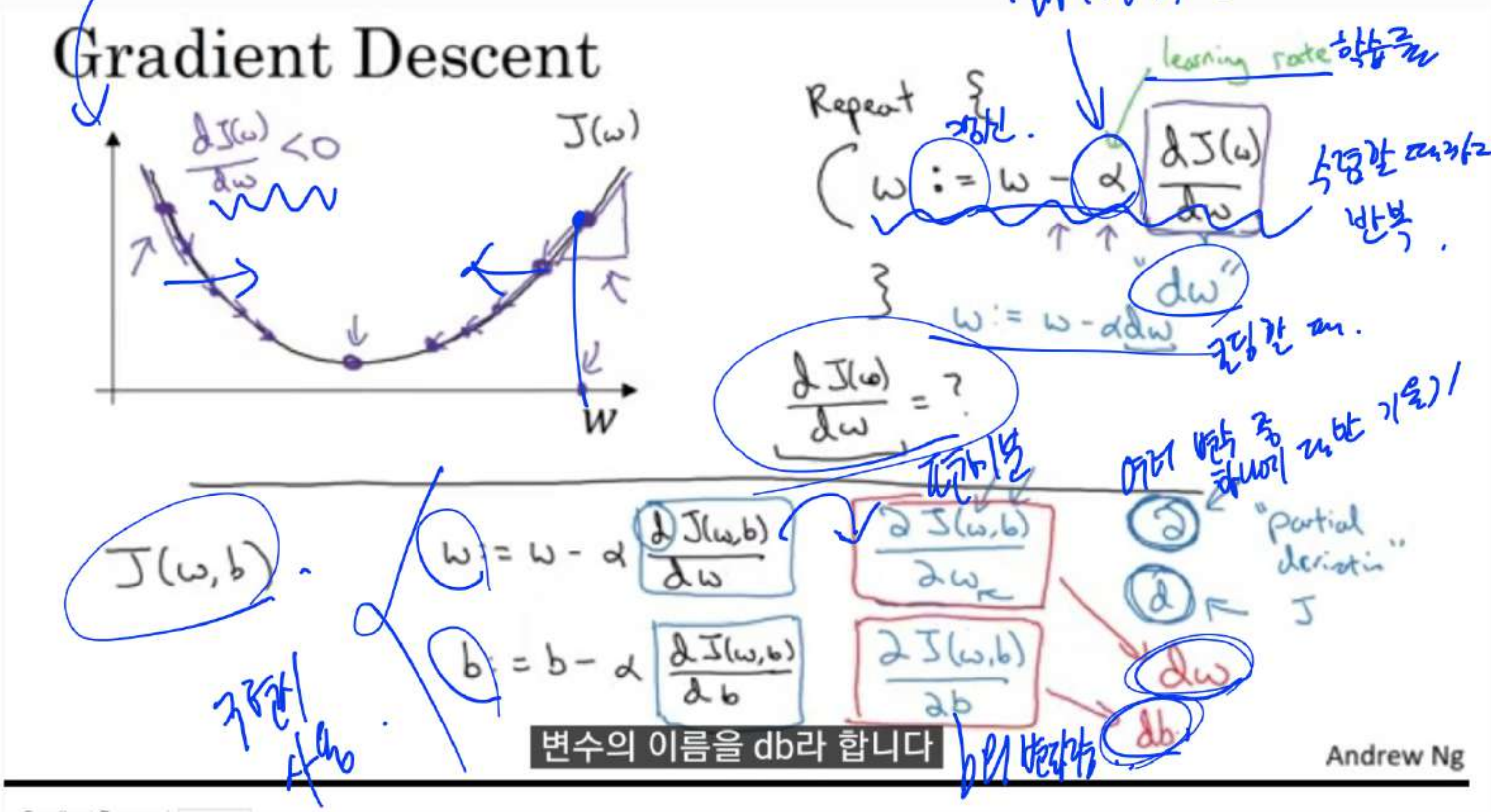
w, b 분리! intercept 값.

40



하나의 실수라고 가정.
다양한 ... 0이라 하자!
가파른 내리막 선택.

b 하나의 값으로 가정.



- 우리의 목표는 실제값(y)에 가까운 예측값($\hat{y}$)를 구하는 것입니다.
- 손실 함수는 하나의 입력특성(x)에 대한 실제값(y)과 예측값($\hat{y}$) 의 오차를 계산하는 함수입니다.
- 보통 손실함수는 $L(\hat{y}, y)=\frac{1}{2}(\hat{y}-y)2$ 식을 사용하지만 로지스틱 회귀에서 이러한 손실함수를 사용하면 지역 최소값에 빠질 수 있기 때문에 사용하지 않습니다. (해당 내용은 향후 다시 나올것이니 걱정하지 않으셔도 됩니다.)
- 로지스틱 회귀에서 사용하는 **손실 함수**는 다음과 같습니다.

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

- 이 함수를 직관적으로 이해하기 위해 두 가지 경우로 나누어 생각해 볼 수 있습니다.

1) y = 0 인 경우 $L(\hat{y},y)=-\log(1-\hat{y})$ 가 0에 가까워지도록 $\hat{y}$ 는 0에 수렴하게 됩니다.
2) y = 1 인 경우 $L(\hat{y},y)=-\log\hat{y}$ 가 0에 가까워지도록 $\hat{y}$ 는 1에 수렴하게 됩니다.

- 하나의 입력에 대한 오차를 계산하는 함수를 손실 함수라고 하며, 모든 입력에 대한 오차를 계산하는 함수는 비용 함수라고 합니다.
- 따라서 **비용 함수**는 모든 입력에 대해 계산한 손실 함수의 평균 값으로 구할 수 있으며 식으로 나타내면 다음과 같습니다.

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

- 항상 기억해야 할 것은, 우리는 실제값과 비슷한 예측값을 원합니다. 즉, 비용 함수의 값이 작아지도록 하는 w와 b를 찾는게 우리의 목표입니다.

- 이전 시간의 비용함수가 전체 데이터셋의 예측이 얼마나 잘 평가되었는지 보는 것이라면, **경사하강법**은 이를 가능케하는 파라미터 w와 b를 찾아내는 방법 중 하나 입니다.
- 우선, 비용 함수는 볼록한 형태여야 합니다. 볼록하지 않은 함수를 쓰게 되면, 경사하강법을 통해 최적의 파라미터를 찾을 수 없습니다.
- 함수의 최소값을 보는 것 때문에, 임의의 점을 골라서 시작합니다.
- 경사하강법은 가장 가파른(steepest) 방향, 즉 함수의 기울기를 따라서 최적의 값으로 한 스텝씩 업데이트하게 됩니다.
- 알고리즘은 아래와 같습니다.

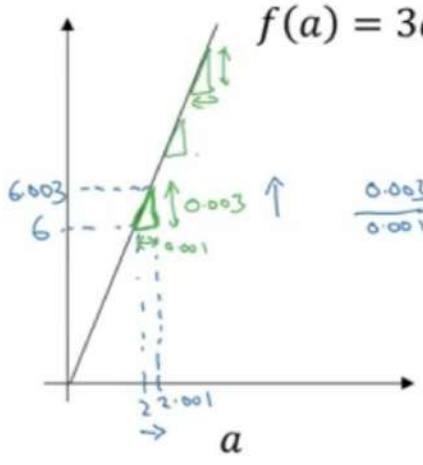
- $w:w-\alpha dwdJ(w,b)$
- $b:b-\alpha dbdJ(w,b)$

- α : 학습률이라고 하며, 얼만큼의 스텝으로 나아갈 것인지 정합니다.
- $dwdJ(w)$

: 도함수라고 하며, 미분을 통해 구한 값 입니다. dw 라고 표기하기도 합니다.

- 만약 $dw>0$ 이면, 파라미터 w 는 기존의 w 값 보다 작은 방향으로 업데이트 될 것이고, 만약 $dw<0$ 이면, 파라미터 w 는 기본의 w 값 보다 큰 방향으로 업데이트 될 것입니다.
- 도함수는 함수의 기울기라고 볼 수 있습니다. 다음 시간에 조금 더 자세히 설명하겠습니다.
- 하나의 변수에 대한 도함수는 $dw=dwdf(w)$ 라고 표기하지만 두 개 이상은 보통 아래와 같이 표현 합니다.
- $dw=\partial w\partial J(w,b)$: 함수의 기울기가 w 방향으로 얼만큼 변했는지 나타냅니다.
- $db=\partial b\partial J(w,b)$: 함수의 기울기가 b 방향으로 얼만큼 변했는지 나타냅니다.

Intuition about derivatives



$a = 2$ $f(a) = 6$

$a = 2.001$ $f(a) = 6.003$

slope (derivative) of $f(a)$ at $a = 2$ is 3

$a = 5$ $f(a) = 15$

$a = 5.001$ $f(a) = 15.003$

slope at $a = 5$ is also 3

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

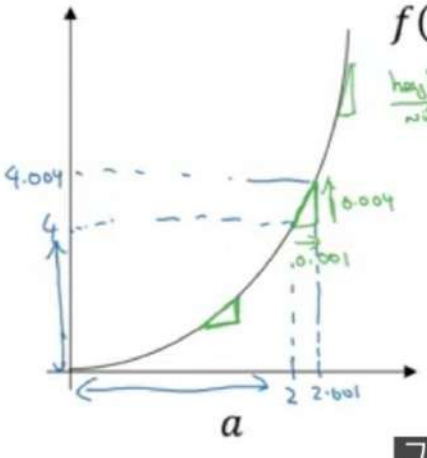
0.001

0.00000001

높이/밑변의 비율은 항상 3:1입니다

Andrew Ng

Intuition about derivatives



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

0.001

$(2a) \times 0.001$

0.001

0.00000....01

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

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

(4.004001)

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

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

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

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

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

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

그래서 f(a)가 증가한 값이 정확히 공식과 같지 않고

Andrew Ng

More derivative examples

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

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

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

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

$f(a) = \log_e(a)$ $\ln(a)$ $\frac{d}{da} f(a) = \frac{1}{a}$

$a = 2$ $f(a) \approx 0.69315$

$a = 2.001$ $f(a) \approx 0.69265$

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

0.0005 0.0005

$\frac{1}{2}$

Andrew Ng

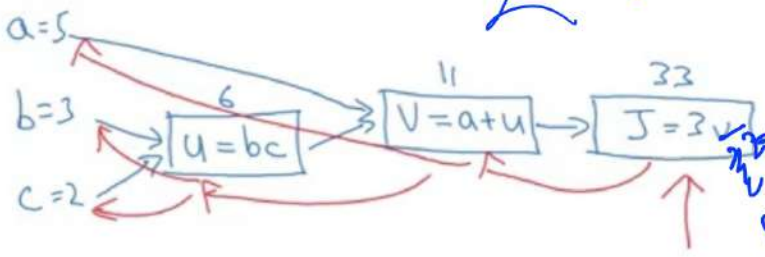
Computation Graph

$J(a,b,c) = 3(a+bc) = 3(5+3\times 2) = 33$

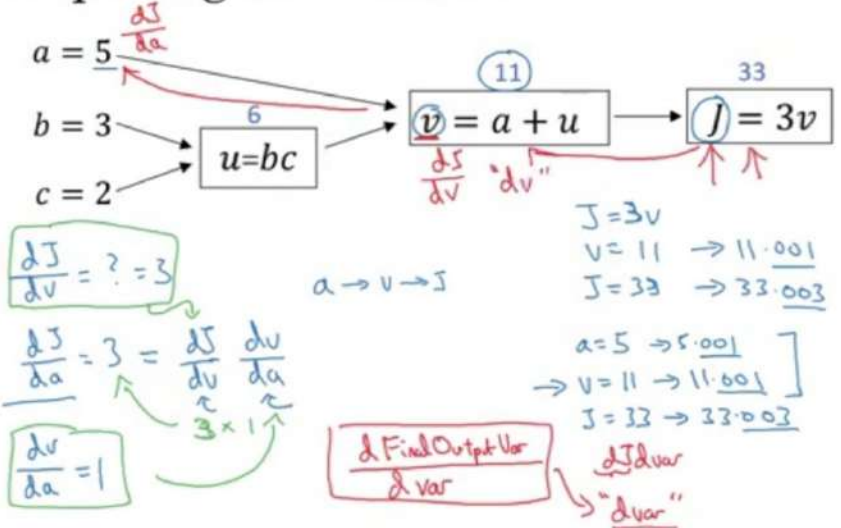
$u = bc$

$v = a+u$

$J = 3v$

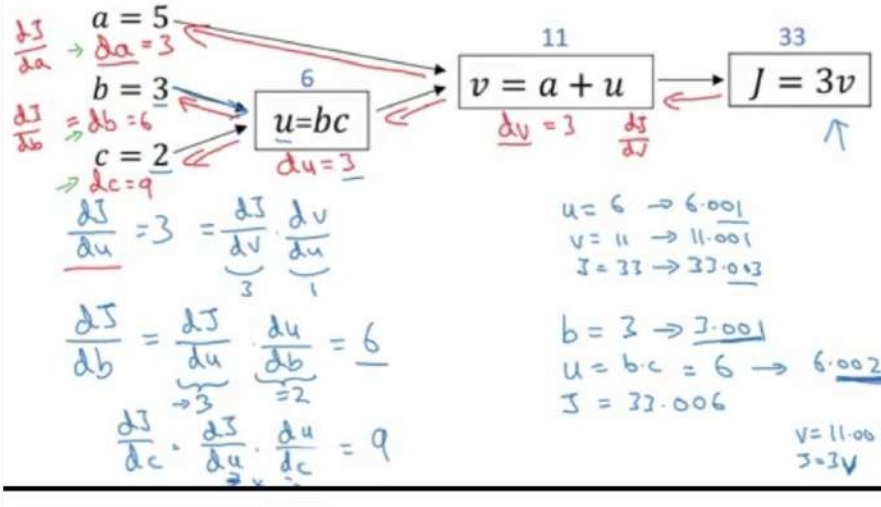


Computing derivatives



Andrew Ng

Computing derivatives



Andrew Ng

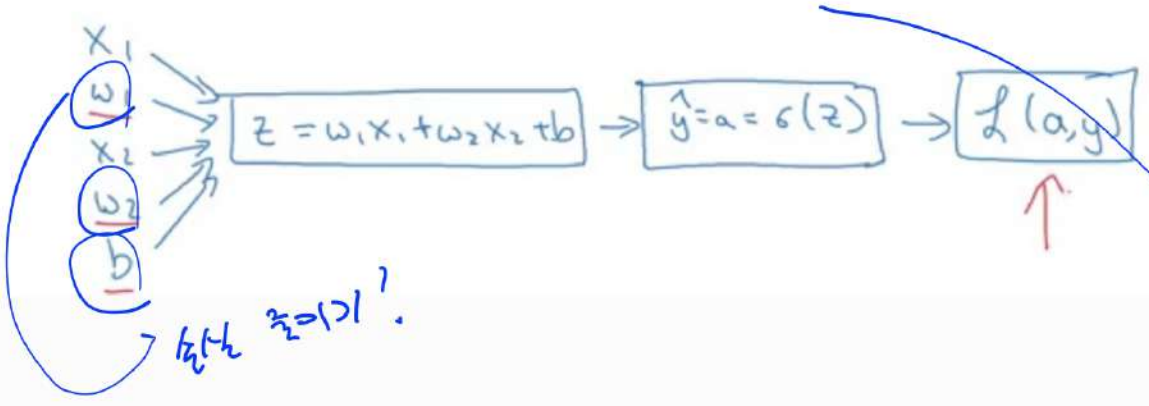
9. 계산그래프를 통해 로지스틱 회귀의 경사하강법

Logistic regression recap

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

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

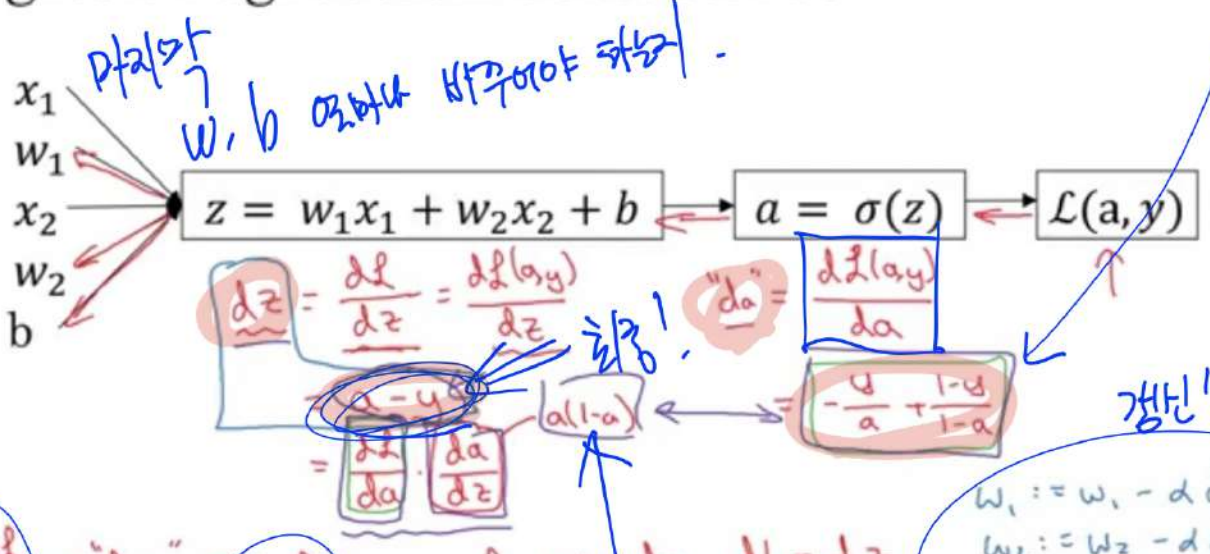
$\rightarrow \mathcal{L}(a, y) = -(y \log(a) + (1-y)\log(1-a))$

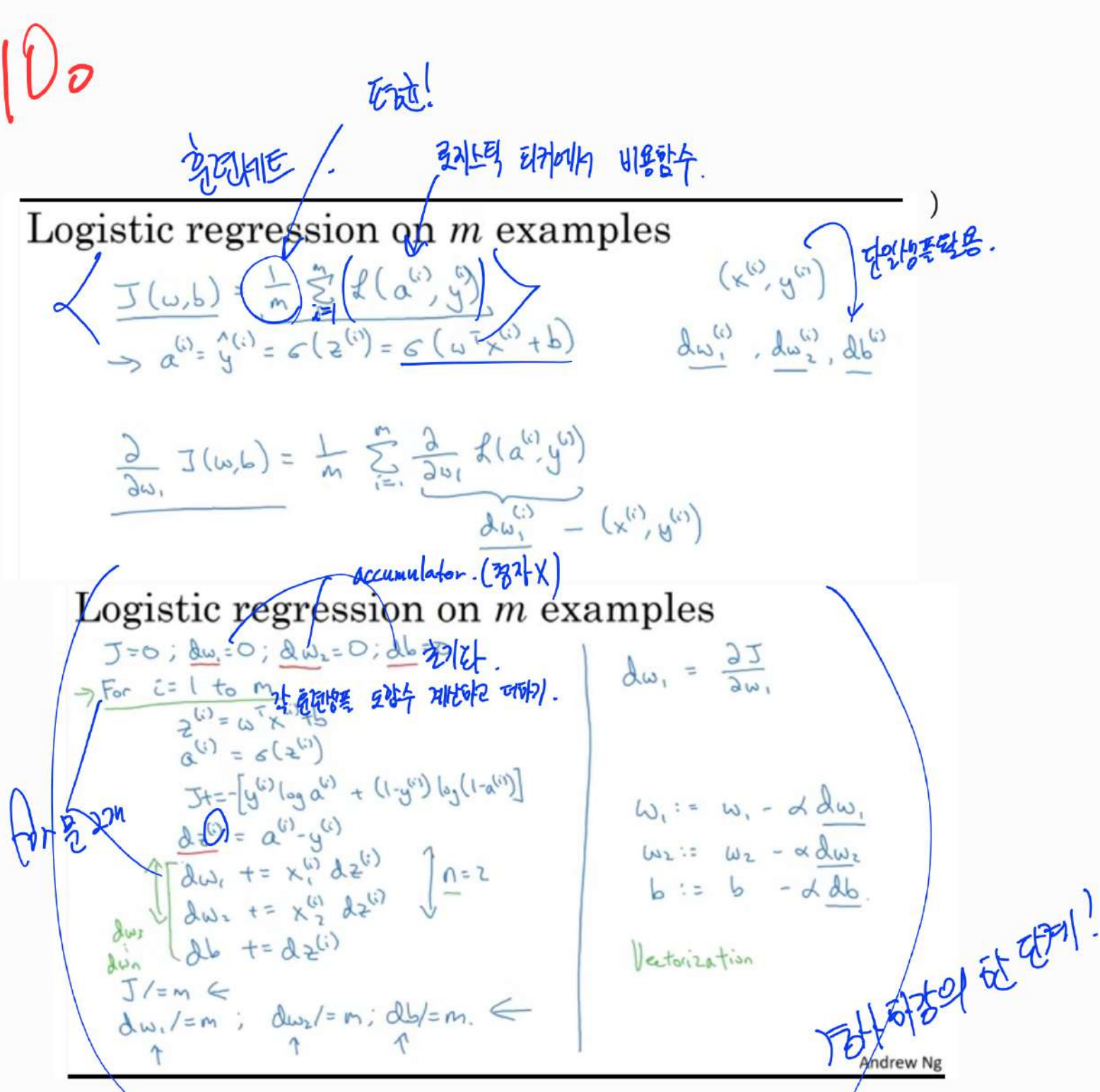


반복

단일 샘플에 대한:

Logistic regression derivatives





- 현재 코드에서는 특성의 개수를 2개로 가정하였지만, 만약 특성의 개수가 많아진다면 이 또한 for문을 이용해 처리해야 합니다. 즉, 이중 for문을 사용하게 되며 이로인해 계산속도가 느려지게 됩니다.
- 다음 시간에는 vectorization을 통해 for문을 사용하지 않고 처리할 수 있는 방법을 배우겠습니다.