2. 신경망과 로지스틱 회귀

이진분류

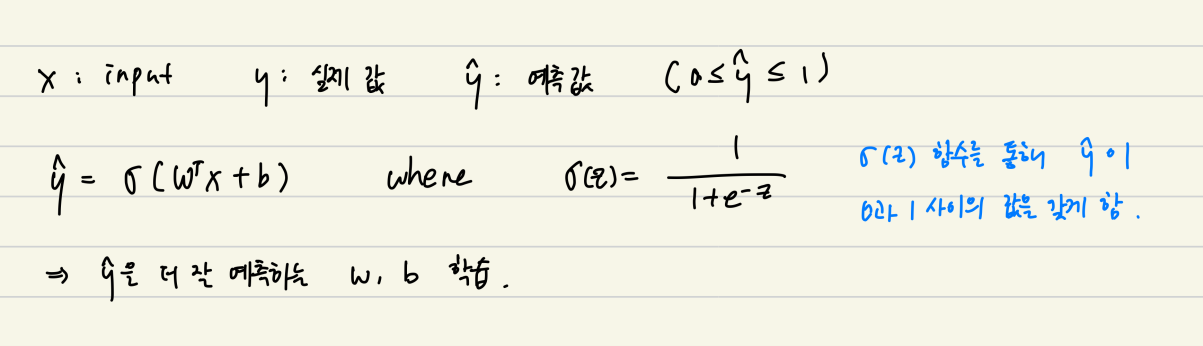
1. 개념

- input x를 1 또는 0으로 분류
- logistic regression란 이진 분류를 하기 위해 사용되는 알고리즘임

로지스틱 회귀

Logistic Regression:

이진 분류 문제에 사용되는 알고리즘



Cost function

Loss function

- input x에 대한 실제값과 예측값의 오차를 계산
- mean squared error는 local optimal에 빠질 수 있으므로 사용하지 않음

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

if $y=1$: $L(\hat{y}, y) = -\log \hat{y} \rightarrow \hat{y} \approx 1$

$y=0$: $L(\hat{y}, y) = -\log(1-\hat{y}) \rightarrow \hat{y} \approx 0$

Cost function

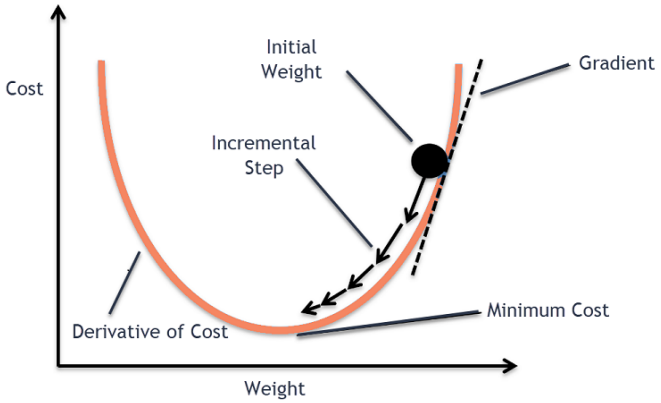
$$J(w, b) = \frac{1}{m} \sum_{i=1}^{m} L(\hat{y}^{(i)}, y^{(i)})$$

Gradient Descent

- logistic regression에서 cost function J(w, b)를 최소화하기 위해 gradient descent 방법을 쓴다
- steps

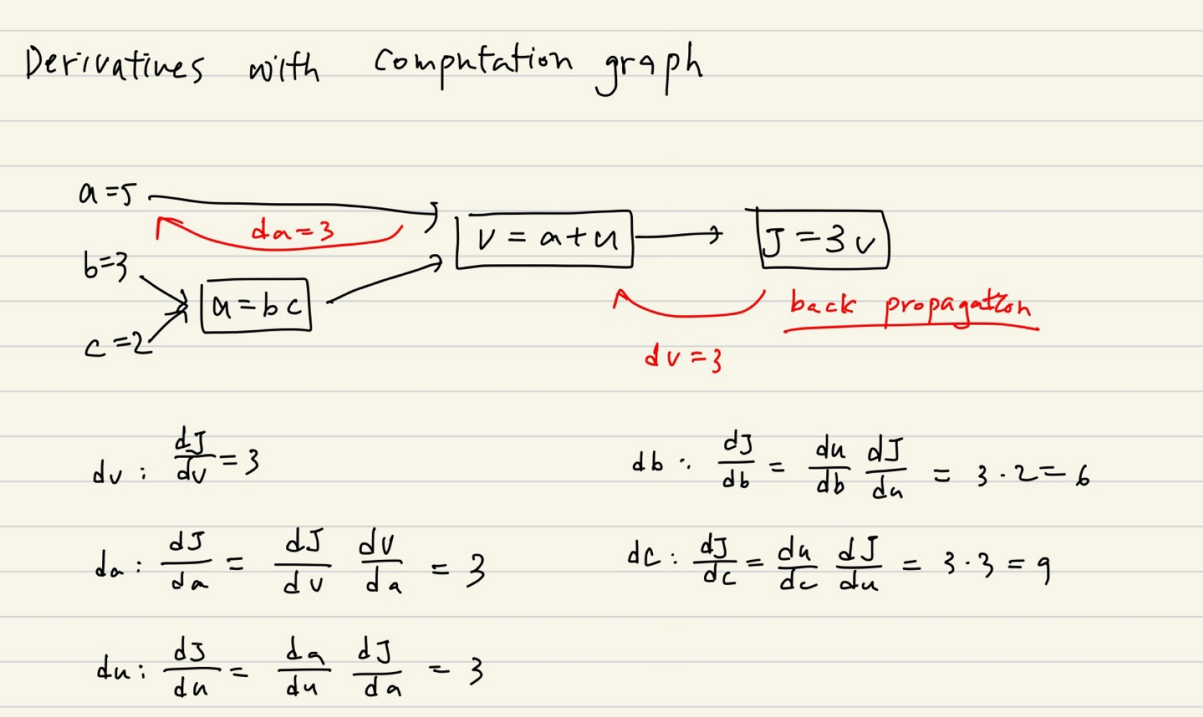
  1) initialize w, b: 주로 0으로 초기화 함

  2) 가장 가파른 방향으로 w, b를 update



Derivatives with Computation Graph

- computation graph에서 back propagation을 통해 derivatives를 구할 수 있음



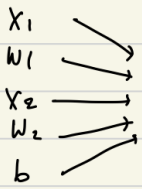
Logistic Regression Gradient Descent

- gradient descent으로 loss function L(a, y)를 최소화하는 파라미터 w, b를 찾음
- 하나의 x에 대한 경사하강법

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

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

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



$z = w_1 x_1 + w_2 x_2 + b$ $\rightarrow$ $\hat{y} = a = \sigma(z)$ $\rightarrow$ $\mathcal{L}(a, y)$

"da" $\frac{d\mathcal{L}}{da}$

$= -\frac{y}{a} + \frac{1-y}{1-a}$

$dz = \frac{d\mathcal{L}}{da}\frac{da}{dz} = a - y$

$dw_1 = x_1 dz$

$dw_2 = x_2 dz$

$db = dz$

$\Rightarrow$ update params.

$w_1 := w_1 - \alpha dw_1$

$w_2 := w_2 - \alpha dw_2$

$b := b - \alpha db$

- m개 샘플의 경사하강법 알고리즘

Gradient descent on m training examples.

$J = 0 ; dw_1 = 0 ; dw_2 = 0, db = 0$

For i=1 to m :

$z^{(i)} = w^T x^{(i)} + b$

$a^{(i)} = \sigma(z^{(i)})$

$J \mathrel{+}= -[y^{(i)} \log a^{(i)} + (1-y^{(i)}) \log(1-a^{(i)})]$

$dz^{(i)} = a^{(i)} - y^{(i)}$

$dw_1 \mathrel{+}= w_1^{(i)} + dz^{(i)}$

$dw_2 \mathrel{+}= w_2^{(i)} + dz^{(i)}$

$db \mathrel{+}= dz^{(i)}$

$J \mathrel{/}= m$

$dw_1 \mathrel{/}= m ; dw_2 \mathrel{/}= m ; db \mathrel{/}= m$

$w_1 := w_1 - \alpha dw_1$

$w_2 := w_2 - \alpha dw_2$

$b := b - \alpha db$

For문 비효율적 → 벡터화

- 두 개의 for 문
  - m개의 샘플에 대한 반복문
  - 파라미터 w1, w2, …, wn에 대한 반복문
- for문은 비효율적이라 벡터화를 사용함