- 로지스틱 회귀는 단일 뉴런 신경망..

순방향 전파

<Back propogation>

cost function $J(\hat{y}, y) = \frac{1}{m}\sum_{i=1}^{m} L^{(i)}(\hat{y}, y)$ with $L^{(i)} = -[y^{(i)}\log\hat{y}^{(i)} + (1-y^{(i)})\log(1-\hat{y}^{(i)})]$

Update $w^{[l]} = w^{[l]} - \alpha \frac{\partial J}{\partial w^{[l]}}$

→ 계산해보자!

→ 차원은.

Improving NNs

A) 다른 종류의 Activation function ex) sigmoid, ReLU, tanh

장점: Easy, 확률적 접근
단점: 양 끝 포화영역에서는 update가 느림.

$$\mathrm{ReLU}(z) = \begin{cases} 0 & z \le 0 \\ z & z > 0 \end{cases}, \quad \mathrm{ReLU}'(z) = 1\{z>0\}$$

$$\tanh = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}, \quad \tanh'(z) = 1 - \tanh(z)^2$$

자주 사용하는 이유?

기울기 소실 문제 X
기존 activation function 보다 빠르다

단점) 0에서 미분 X, 음수 → 0으로 assume

* Why do we need activation function.

$$\hat{y} = a^{[3]} = z^{[3]} = w^{[3]} a^{[2]} + b^{[3]}$$

$$= Wx + B \quad \text{with } W = W^{[3]} W^{[2]} W^{[1]}$$

$$B = W^{[3]} W^{[2]} b^{[1]} + W^{[3]} b^{[2]} + b^{[3]}$$

비선형적 요소 포함시키기 위해서 응용 → Network의 복잡성 activation function.

B) Initialization methods

격차 너무 크거나, 작음 → 정규화!

Normalizing your input

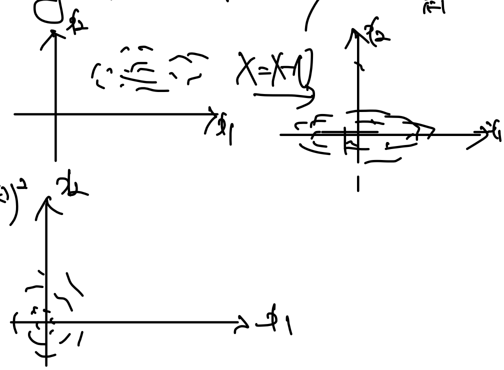
$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$

$\mu = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}$

$x = x - \mu$

$\sigma^2 = \frac{1}{m}\sum_{i=1}^{m} (x^{(i)})^2$

$x = x / \sigma$



μ, σ는 train, test 에서 동일!

Vanishing / Exploding gradients

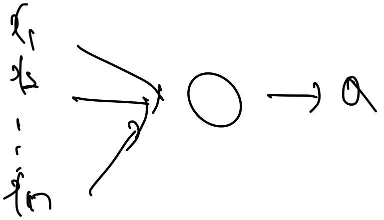
$\hat{y} = W^{[L]} W^{[L-1]} \cdots W^{[1]} X + B.$

2가지경우 (i) $W^{[L]} = \begin{pmatrix} 1.5 & 0 \\ 0 & 1.5 \end{pmatrix}$ assume $\rightarrow \hat{y} = \begin{pmatrix} 1.5^L & 0 \\ 0 & 1.5^L \end{pmatrix} x$

Explode 해버림.

(ii) $W^{[L]} = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix}$ 〃 → Vanish

Example with 1 neuron



$a = \sigma(z)$

$z = w_1 x_1 + \cdots + w_n x_n$

large n → small w_i

$w_i \sim \frac{1}{n}$ 로 초기화함..

i) $W_i^{[l]}$ = np.random.randn(shape) * np.sqrt($\frac{1}{n^{[l-1]}}$)

↳ 대칭문제 방지.

들어오는 입력들에 비례하여 조절 → sigmoid에서 잘 작동함.

ReLU에선 1 → 2로 바꾸자!

ii) Xavier Initialization

$$W^{[l]} \sim \sqrt{\frac{1}{n^{[l-1]}}}$$

for tanh

iii) He Initialization

$$W^{[l]} \sim \sqrt{\frac{2}{n^{[l]} + n^{[l-1]}}}$$

← 역전파도 신경써줌, for ReLU

c) optimization

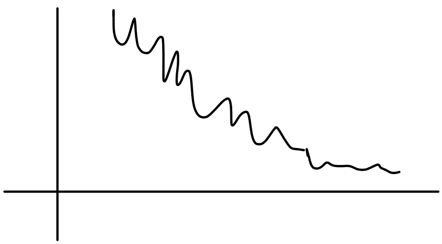
Batch ⟷ Stochastic , 절충안 Mini batch.

→ Algorithm 작동방식

for iteration , select batch , Forward prop , Backward prop

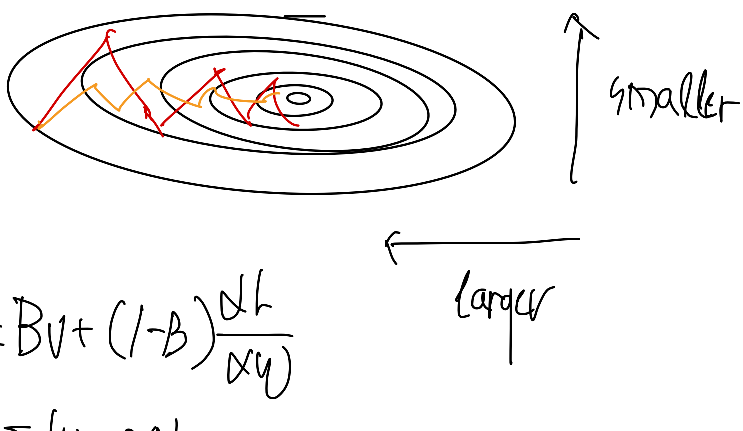
⇒ update $w^{[l]}$, $b^{[l]}$

graph



* Momentum Algorithm

:: 진행을 가속화. . - -



$$\begin{cases} v = \beta v + (1-\beta)\dfrac{\partial L}{\partial w} \\ w = w - \alpha v \end{cases}$$