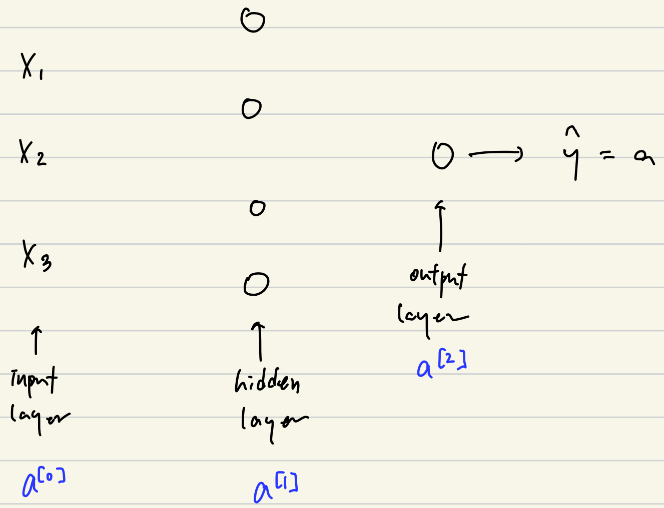
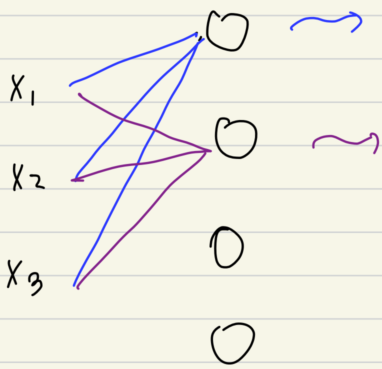
NN 구성





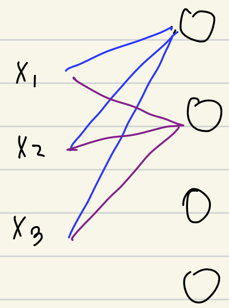
$$\begin{cases} z_1^{[1]} = w_1^{[1]T} x + b \\ a_1^{[1]} = \sigma(z_1^{[1]}) \end{cases}$$

$$\begin{cases} z_2^{[1]} = w_2^{[1]} x + b \\ a_2^{[1]} = \sigma(z_2^{[1]}) \end{cases}$$

$\vdots$

- 벡터화

- layer 1



$$\begin{cases} z_1^{[1]} = w_1^{[1]T} x + b \\ a_1^{[1]} = \sigma(z_1^{[1]}) \end{cases}$$

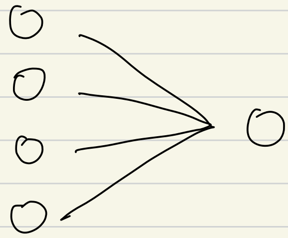
$$\begin{cases} z_2^{[1]} = w_2^{[1]} x + b \\ a_2^{[1]} = \sigma(z_2^{[1]}) \end{cases}$$

$$\vdots$$

$$z^{[1]} = \underset{(4,3)}{\begin{bmatrix} w_1^{[1]T} \\ w_2^{[1]T} \\ w_3^{[1]T} \\ w_4^{[1]T} \end{bmatrix}} \underset{(3,1)}{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}} + \underset{(4,1)}{\begin{bmatrix} b_1^{[1]} \\ b_2^{[1]} \\ b_3^{[1]} \\ b_4^{[1]} \end{bmatrix}} = \begin{bmatrix} w_1^{[1]T} x + b_1^{[1]} \\ \vdots \\ w_4^{[1]T} x + b_4^{[1]} \end{bmatrix} = \begin{bmatrix} z_1^{[1]} \\ \vdots \\ z_4^{[1]} \end{bmatrix}$$

$$a^{[1]} = \begin{bmatrix} \sigma(z_1^{[1]}) \\ \vdots \\ \sigma(z_4^{[1]}) \end{bmatrix} = \underset{(4,1)}{\begin{bmatrix} a_1^{[1]} \\ \vdots \\ a_4^{[1]} \end{bmatrix}}$$

- layer 2



$$W^T = \underset{(1,4)}{w^{[2]}}, \quad b = \underset{(1,1)}{b^{[2]}}$$

$$z^{[2]} = w^{[2]} a^{[1]} + b^{[2]}$$

$$\underset{(1,1)}{a^{[2]}} = \sigma(z^{[2]})$$

- Vectoring across multiple examples

* $a^{[n](i)}$ n: layer, i: example

- 한 개의 샘플에 대한 벡터화

$$\begin{cases} z^{[1]} = W^{[1]}x + b^{[1]} \\ a^{[1]} = \sigma(z^{[1]}) \\ z^{[2]} = W^{[2]}a^{[1]} + b^{[2]} \\ a^{[2]} = \sigma(z^{[2]}) \end{cases} \Rightarrow$$

- m개의 샘플

for i = 1 to m,

$$\begin{cases} z^{[1](i)} = W^{[1](i)}x^{(i)} + b^{[1]} \\ a^{[1](i)} = \sigma(z^{[1](i)}) \\ z^{[2](i)} = W^{[2](i)}a^{[1](i)} + b^{[2]} \\ a^{[2](i)} = \sigma(z^{[2](i)}) \end{cases}$$

- m개의 samples에 대한 X, Z, A

$$X = \begin{bmatrix} x^{(1)} & x^{(2)} & \cdots & x^{(m)} \end{bmatrix} \quad (n_x, m)$$

(m : # training samples)

$$Z^{[1]} = \overbrace{\begin{bmatrix} z^{[1](1)} & z^{[1](2)} & \cdots & z^{[1](m)} \end{bmatrix}}^{\#\text{ training samples}} \updownarrow \#\text{ hidden units}$$

$$A^{[1]} = \begin{bmatrix} a^{[1](1)} & a^{[1](2)} & \cdots & a^{[1](m)} \end{bmatrix}$$

$$\Rightarrow \begin{cases} Z^{[1]} = W^{[1]}X + b^{[1]} \\ A^{[1]} = \sigma(Z^{[1]}) \\ Z^{[2]} = W^{[2]}A^{[1]} + b^{[2]} \\ A^{[2]} = \sigma(Z^{[2]}) \end{cases}$$

- Vectorized implementation

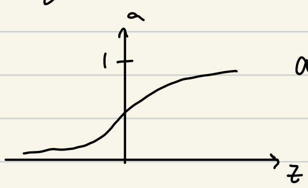
$$Z^{[1]} = \left[W^{[1]}x^{(1)} + b^{[1]} \quad W^{[1]}x^{(2)} + b^{(1)} \quad \cdots \quad W^{[1]}x^{(m)} + b^{[1]} \right]$$

$$= W^{[1]} \left[x^{(1)} \; x^{(2)} \cdots x^{(m)} \right] + b^{[1]}$$

$$= W^{[1]} X + b^{[1]}$$

< Activation functions >

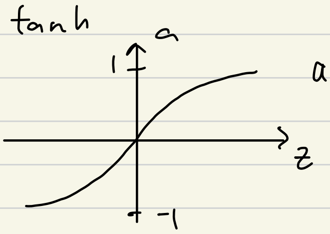
- Sigmoid function



$a = \frac{1}{1+e^{-z}}$

← Never use sigmoid except output layer in binary classification !

- tanh



$a = \tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$

- 은닉층 : tanh > sigmoid ← 값이 ±1 사이라서 평균이 0에 가깝기 때문.

- 출력층 : sigmoid ← 값이 0~1 사이라서 이진분류에 유리함

- sigmoid, tanh 단점
  - z가 매우 크거나 작을 때 기울기가 0에 가까워지면서 경사하강법 느려짐

- ==ReLu==

$a = \max(0, z)$

- 단점 : $z<0$ 일때 기울기가 0

↓

Leaky ReLu : $z<0$ 일때도 약간의 기울기를 줌

- 장점 : 대부분의 z에서 기울기가 0보다 큼.

⇒ 이진분류 출력층 → sigmoid
다른 경우 → ReLu

<신경망 네트워크와 경사하강법>

Parameters : $w^{[1]}$, $b^{[1]}$, $w^{[2]}$, $b^{[2]}$

dim : $(n^{[1]}, n^{[0]})$, $(n^{[1]}, 1)$, $(n^{[2]}, n^{[1]})$, $(n^{[2]}, 1)$

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

Gradient descent :

Repeat {

Compute Params $(\hat{y}^{(i)}, i = 1, \cdots, m)$

$dw^{[1]} = \frac{dJ}{dw^{[1]}}$, $db^{[1]} = \frac{dJ}{db^{[1]}}$, ... # Params 의 도함수 구함

$W^{[1]} := w^{[1]} - \alpha dW^{[1]}$ # update params

$b^{[1]} := b^{[1]} - \alpha dw^{[1]}$

⋮

} 수렴할 때까지 반복

<Forward propagation>

$$Z^{[1]} = W^{[1]}X + b^{[1]}$$

$$A^{[1]} = g^{[1]}(Z^{[1]})$$

$$Z^{[2]} = W^{[2]}A^{[1]} + b^{[2]}$$

$$A^{[2]} = g^{[2]}(Z^{[2]}) = \sigma(Z^{[2]})$$

<Backward propagation>

$$dZ^{[2]} = A^{[2]} - Y$$

$$dW^{[2]} = \frac{1}{m} dZ^{[2]} A^{[1]T}$$

$\cancel{(n^{[2]},\ )} \to (n^{[2]}, 1)$

$$db^{[2]} = \frac{1}{m} \text{np.sum}(dZ^{[2]}, \text{axis}=1, \text{keepdims}=\text{True})$$

$$dZ^{[1]} = W^{[2]T} dZ^{[2]} * g^{[1]\prime}(Z^{[1]})$$

element-wise product

$$dW^{[1]} = \frac{1}{m} dZ^{[1]} X^T$$

$$db^{[1]} = \frac{1}{m} \text{np.sum}(dZ^{[1]}, \text{axis}=1, \text{keepdims}=\text{True})$$

<역전파 도함수 계산식 유도 과정>

$$\begin{tikzcd} x \arrow[dr] & & & \\ w \arrow[r] & z = w^T x + b \arrow[r, shift left] \arrow[l, shift left, "3"] & a = \sigma(z) \arrow[r, shift left] \arrow[l, shift left, "2"] & L(a,y) \arrow[l, shift left, "1"] \\ b \arrow[ur] & & & \end{tikzcd}$$

① $da = \frac{d}{da} L(a,y) = -y\log a - (1-y)\log(1-a) = -\frac{y}{a} + \frac{1-y}{1-a}$

② $dz = \frac{dL}{dz} = \frac{da}{dz} \cdot \frac{dL}{da}$

$= \frac{d}{dz} g(z) \cdot da \qquad (a = g(z) = \sigma(z))$

$= da \cdot g'(z)$

$= a - y$

③ $dw = dz \cdot x$

$db = dz$

$$x,\ W^{[1]},\ b^{[1]} \underset{④}{\longrightarrow} z^{[1]} = W^{[1]}x + b^{[1]} \underset{③}{\longrightarrow} a^{[1]} = \sigma(z^{[1]})$$

$$a^{[1]},\ W^{[2]},\ b^{[2]} \underset{②}{\longrightarrow} z^{[2]} = W^{[2]}x + b^{[2]} \underset{①}{\longrightarrow} a^{[2]} = \sigma(z^{[2]}) \longrightarrow L(a^{[2]}, y)$$

① $dz^{[2]} = a^{[2]} - y$

② $dw^{[2]} = dz^{[2]} a^{[1]T}$

$db^{[2]} = dz^{[2]}$

③ $dz^{[1]} = w^{[2]T} dz^{[2]} * g^{[1]\prime}(z^{[1]})$

* dim: $w^{[2]}$ $(n^{[2]}, n^{[1]}) = (1, n^{[1]})$

$z^{[2]}, dz^{[2]}$ $(n^{[2]}, 1) = (1, 1)$

$z^{[1]}, dz^{[1]}$ $(n^{[1]}, 1)$

$$\underset{(n^{[1]},1)}{dz^{[1]}} = \underset{(n^{[1]},n^{[2]})}{w^{[2]T}} \ \underset{(n^{[2]},1)}{dz^{[2]}} * \underset{(n^{[1]},1)}{g^{[1]\prime}(z^{[1]})}$$

④ $dw^{[1]} = dz^{[1]} \cdot x^T$, $x^T = a^{[0]T}$

$db^{[1]} = dz^{[1]}$

$$dz^{[2]} = a^{[2]} - y$$

$$dW^{[2]} = dz^{[2]} a^{[1]^T}$$

$$db^{[2]} = dz^{[2]}$$

$$dz^{[1]} = W^{[2]T} dz^{[2]} * g^{[1]\prime}(z^{[1]})$$

$$dW^{[1]} = dz^{[1]} x^T$$

$$db^{[1]} = dz^{[1]}$$

- 벡터화

$$dZ^{[2]} = A^{[2]} - Y$$

$$dW^{[2]} = \frac{1}{m} dZ^{[2]} A^{[1]T}$$

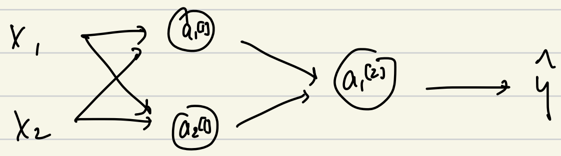
$$db^{[2]} = \frac{1}{m} np.sum(dZ^{[2]}, axis=1, keepdims=True)$$

$$dZ^{[1]} = W^{[2]T} dZ^{[2]} * g^{[1]\prime}(Z^{[1]})$$

$$dW^{[1]} = \frac{1}{m} dZ^{[1]} X^T$$

$$db^{[1]} = \frac{1}{m} np.sum(dZ^{[1]}, axis=1, keepdims=True)$$

<랜덤 초기화>



• 파라미터를 0으로 초기화하는 경우

$W^{[1]} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$ $\quad b^{[1]} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

$a_1^{[1]} = a_2^{[1]}$ $\quad dz_1^{[1]} = dz_1^{[2]}$

⇒ $a_1^{[1]}$, $a_2^{[1]}$ hidden unit이 항상 같은 계산을 함.

⇒ hidden unit이 하나인 것과 같음.

⇒ 랜덤하게 파라미터를 초기화해야함

• Random initialization

$W^{[1]}$ = np.random.rand((2,2)) * 0.01

→ 작은 수로 초기화하는 이유?

⇒ W가 커지면 z도 커짐

⇒ sigmoid, tanh 에서는 z가 커질수록 기울기가 0에 가까워지기 때문에 학습 속도가 느려짐.

$b^{[1]}$ = np.zeros((2,1))

$W^{[2]}$ = np.random.rand((1,2)) * 0.01

$b^{[2]} = 0$