Activation function

Khyeyoon

Q. 왜 비선형함수를 사용할까?

선형함수를 사용한다면 여러개의 은닉층을 쌓아도 기대하는 결과를 얻을 수 없음

→ 여러개의 은닉층을 사용하는 것이나 하나의 은닉층을 사용하는 것이나 다를 바 없음

$$g(z) = z \quad = \quad g(g(g(z))) = z$$

Khyeyoon

Formulas for computing derivatives

Propagation :

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

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

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

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

Back propagation :

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

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

1차원 방지
$(n^{[2]},) \rightarrow (n^{[2]}, 1)$

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

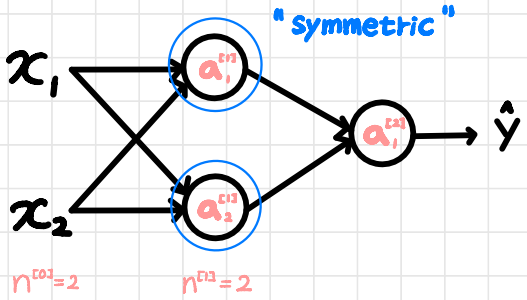
$$dZ^{[1]} = W^{[2]T} dZ^{[2]} * g^{[1]\prime}(Z^{[1]})$$

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

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

Khyeyoon

What happens if you initialize weights to zero?



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

→ $a_1^{[1]} = a_2^{[1]}$

→ $dz_1^{[1]} = dz_2^{[1]}$

Hidden unit이 동일한 연산을 수행

$$W^{[1]} = \begin{bmatrix} = \\ = \end{bmatrix}$$

→ Unit이 하나인 것과 마찬가지

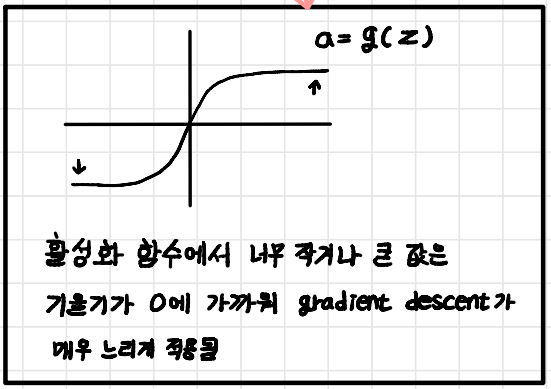
Random initialization

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

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

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

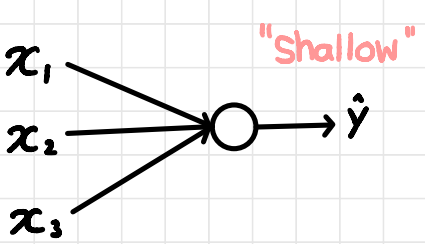
$b^{[2]} = 0$



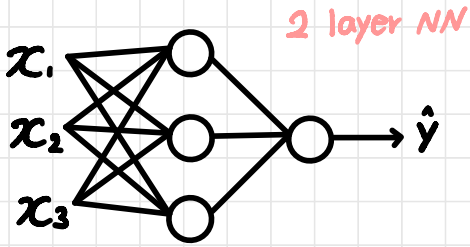
활성화 함수에서 너무 작거나 큰 값은 기울기가 0에 가까워 gradient descent가 매우 느리게 작용됨

Khyeyoon

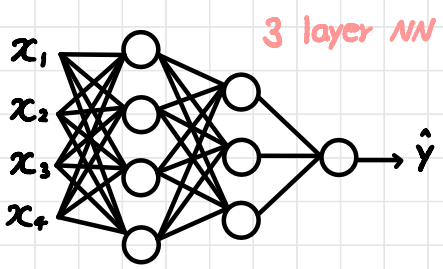
What is a deep neural network ?



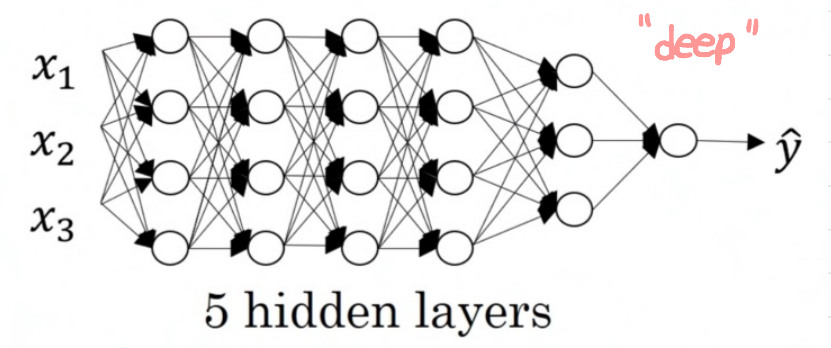
logistic regression



1 hidden layer

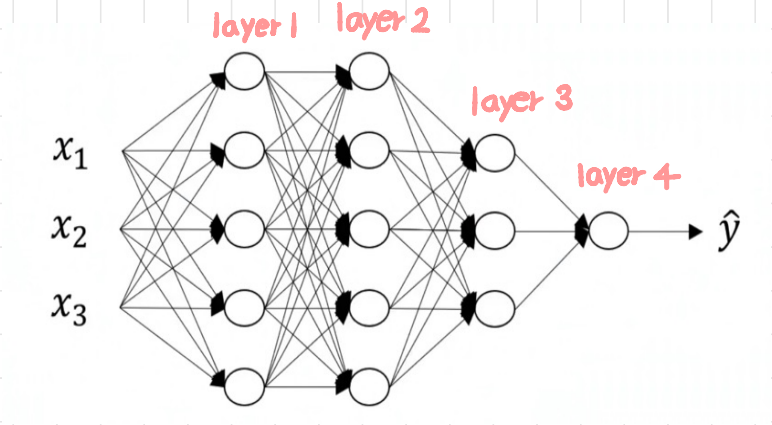


2 hidden layers



Khyeoon

Deep neural network notation



$L = 4$ (Layer 4)

$n^{[l]}$ = layer l의 Units

→ $n^{[1]} = 5$, $n^{[2]} = 5$, $n^{[3]} = 3$, $n^{[4]} = n^{[L]} = 1$

$n^{[0]} = n_x = 3$

$a^{[l]}$ = layer l의 Activation

$= g^{[l]}(z^{[l]})$