Random Initialization

미래연구소 14기 4주차

a가 symmetric이면
dz가 symmetric인 이유는
뒷 장의 식 때문입니다.

1. zero initialization의 문제점 1) row symmetric

$$W^{[l]} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

W[l]이 모두 0이면,
g(W[l]X + b) = A[l]의 행들의 원소가 모두 같은 row symmetric한 현상이 생깁니다.

$$a_1^{[l]} = a_2^{[l]}$$ **row symmetric**

(column은 b를 다르게 설정하면 symmetric 깰 수 있지만 row symmetric은 안 깨집니다.)

$$dz_1^{[l]} = dz_2^{[l]}$$ **row symmetric**

$$dW^{[l]} = \frac{1}{m} dZ^{[l]} A^{[l-1]T}$$ **row symmetric**

$$W^{[l]} = W^{[l]} - \alpha dW^{[l]}$$ **row symmetric**

node를 여러개 설정했음에도

node가 1개인 효과를 갖는다.

1. zero initialization의 문제점 2) vanishing gradient

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

$$W^{[l]} = W^{[l]} - \alpha dW^{[l]}$$

$$dZ^{[l]} = dA^{[l]} * g'(Z^{[l]})$$

$$dA^{[l-1]} = \frac{\partial \mathcal{L}}{\partial A^{[l-1]}} = W^{[l]T} dZ^{[l]}$$

왼쪽의 식들은 W4 L6에 나와있는 내용이니 증명을 생략합니다.
(증명하지 않아도 됩니다.)

W[l] = 0이 되면,

dW[l] = 0이 되는 문제가 생긴다.

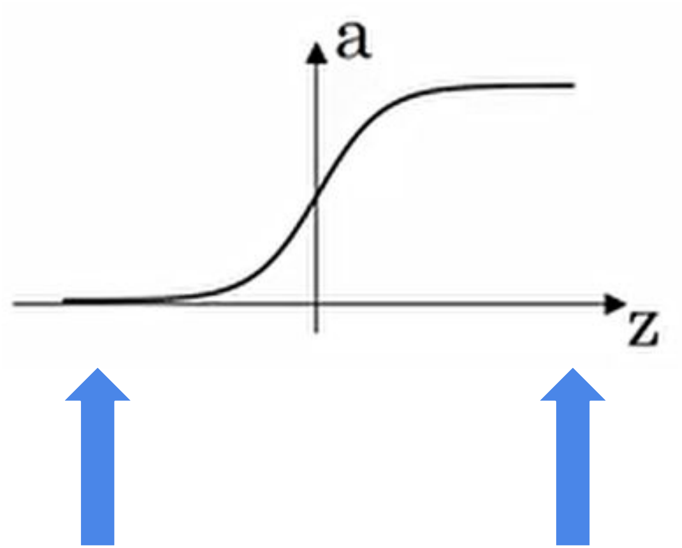
(정확히는 W[l+1] =0 => dW[l] = 0)

2. 대책 1)np.random.randn

$$W^{[l]} = np.random.randn(shape)$$

W[l]이 row symmetric 되는 것을 막는다.

2. 대책 2) 0이 아닌 적절한 상수 곱한다.



1> 0보다는 큰 수를 곱한다.

2> W가 너무 크면 Z가 activation의 양 끝에 위치해
saturation 현상이 발생할 수 있으니
너무 크지 않은 수를 곱한다.

3. 결론

$$W^{[l]} = 0.01 * np.random.randn(shape)$$

재밌는 사이트: https://www.deeplearning.ai/ai-notes/initialization/