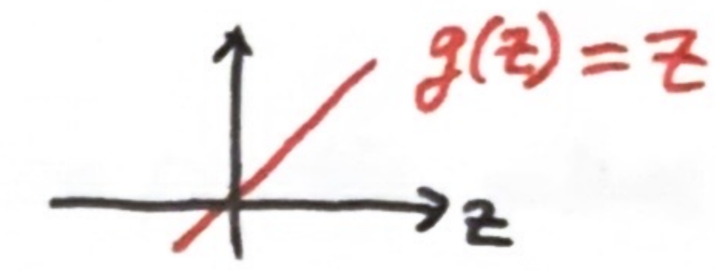
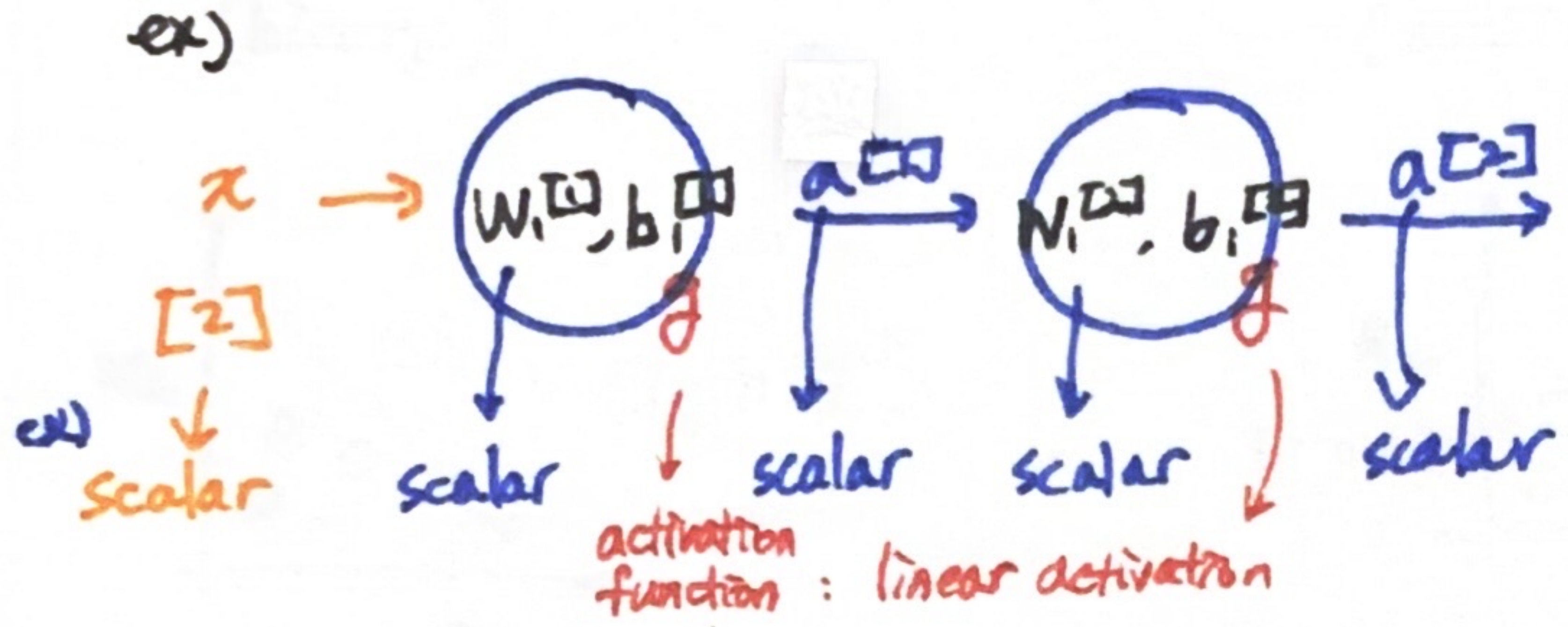
< Why do we need activation function? >

* The reason that we avoid to use 'linear activation function' ⇒ 

ex)



$$g(z) = z \quad \Rightarrow \quad a^{[1]} = g(\overbrace{w_1^{[1]}x + b_1^{[1]}}^{z}) = \overbrace{w_1^{[1]}x + b_1^{[1]}}^{z}$$

$$a^{[2]} = g(w_2^{[2]}a^{[1]} + b_2^{[2]}) = w_1^{[2]} \cdot a^{[1]} + b_2^{[2]}$$

$$= w_1^{[2]}(w_1^{[1]}x + b_1^{[1]}) + b_2^{[2]}$$

$$\therefore \vec{a}^{[2]} = (\underbrace{\vec{w}_1^{[2]}\vec{w}_1^{[1]}}_{W})x + \underbrace{w_1^{[2]}b_1^{[1]} + b_2^{[2]}}_{b}$$

if set

$$= Wx + b$$

↳ 결과가 다시 linear function이 되버림

linear function of linear function is itself a linear function

∴ hidden layer에서는 linear function을 사용하면 안됨

① 신경망의 층을 깊게 하는 것이 의미가 없어짐

ex) 활성화함수 : $h(x) = c(x)$ $(c=상수)$

hidden layer → 3층

$$\Rightarrow y(x) = h(h(h(x))) = c^3x \quad (\text{if } a = c^3)$$

$$= ax$$

= * 선형함수를 여러번 사용하는 것은 결국 마지막에 선형함수 한 번 쓰는 것과 다른게 없다

↳ 은닉층이 없는 네트워크로도 표현할 수 있음

② backpropagation 불가능

- 역전파는 활성화함수를 미분해 이를 이용하여 cost를 줄이는 과정
- 하지만 linear function의 미분값은 입력값 x와 관계없는 상수값
- 따라서 예측값과 가중치에 대한 상호관계 정보는 얻을 수 없음.