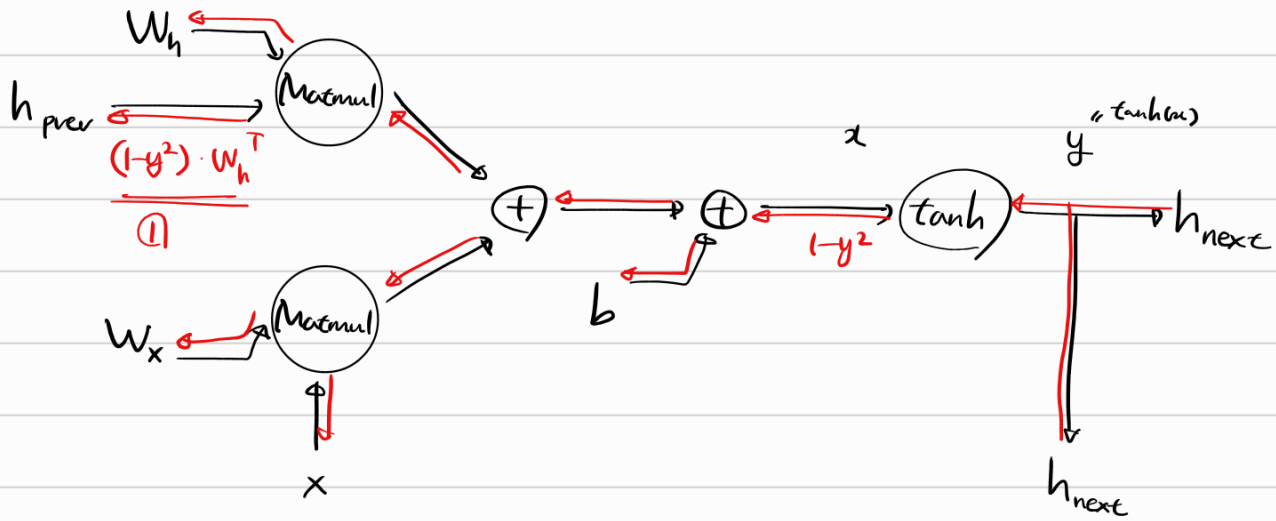
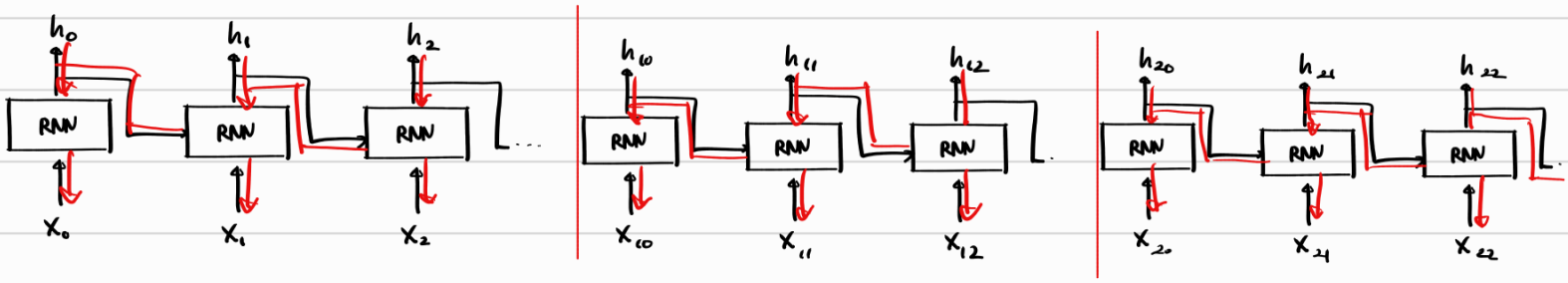
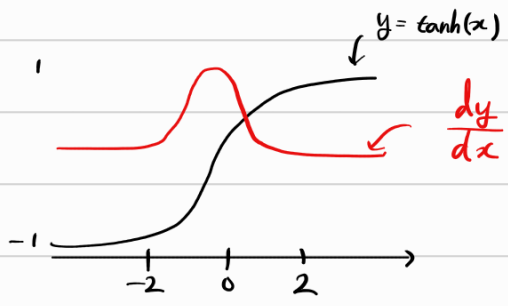
CH. 1 RNN의 한계

Recall. RNN





Thm 1.1 gredient exploding & Vanish on RNN



P1. $dh_{prev} = (1-y^2) \times W_h^T$ 만약 W_h의 특이값이 1보다 크면? or 작으면?

Could admit Nan

P2. $1-y^2$ 은 monotonly decrease.

그리고 W_h^T도 역시 소실을 야기할 수 있다.

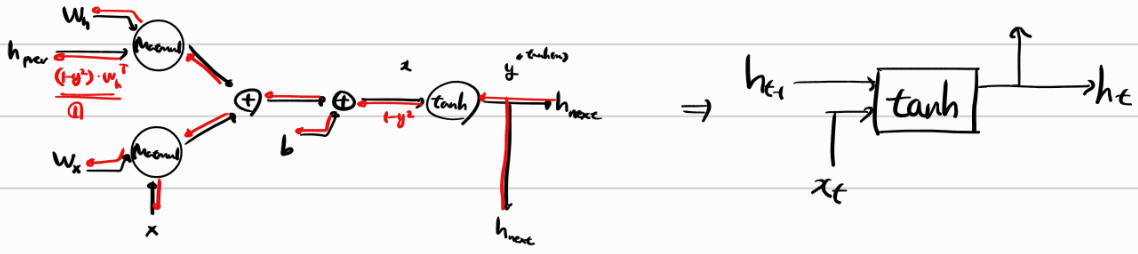
That is, dh_{prev}의 소실, 폭발 가능성이 있다. 이는 장기 기억력에 악영향을 준다!

Thm 1.2 gredients clipping can help gredicare exploding

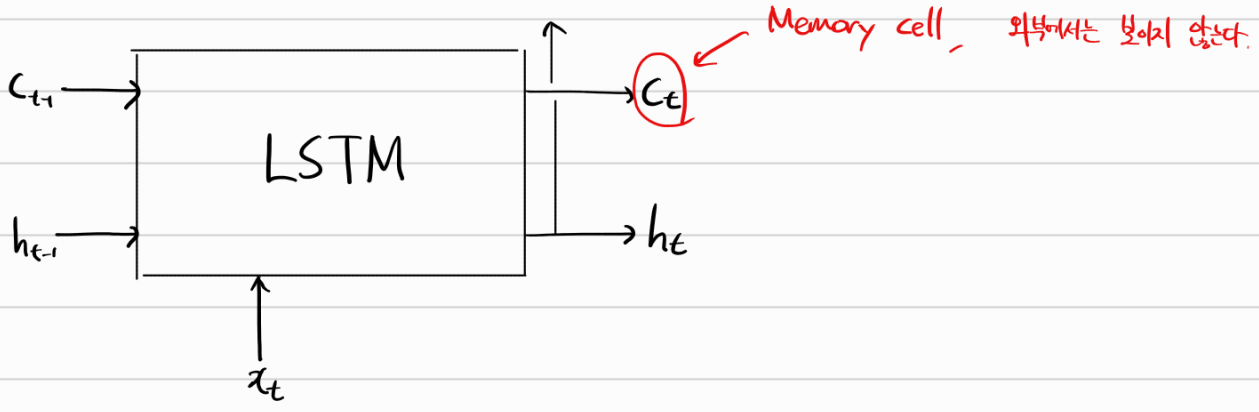
If $\|\hat{g}\| \geq$ threshold, Then $\hat{g} = \frac{\text{Threshold}}{\|\hat{g}\|} \times \hat{g}$

CH. 2 gradients vanish and LSTM

Notation



Def 2.1 LSTM Architecture



Def 2.2 output gate

sigmoid

Let $O = \sigma(x_t W_x^{(o)} + h_{t-1} W_h^{(o)} + b^{(o)})$

0~1, 값이 클수록 정보 반영 ↑

hadamard Product

Then $h_t = O \odot \tanh(c_t)$

Note 2.3 tanh vs sigmoid

tanh: 정보 자체의 강약

sigmoid: 정보 Gate

Def 2.3 forget gate

Let $f = \sigma(x_t W_x^{(f)} + h_{t-1} W_h^{(f)} + b^{(f)})$

Then $C_t = f \odot \tanh(c_{t-1})$

Def 2.4 New memory cell

Let $g = \tanh\left(x_t W_x^{(g)} + h_{t-1} W_h^{(g)} + b^{(g)}\right)$

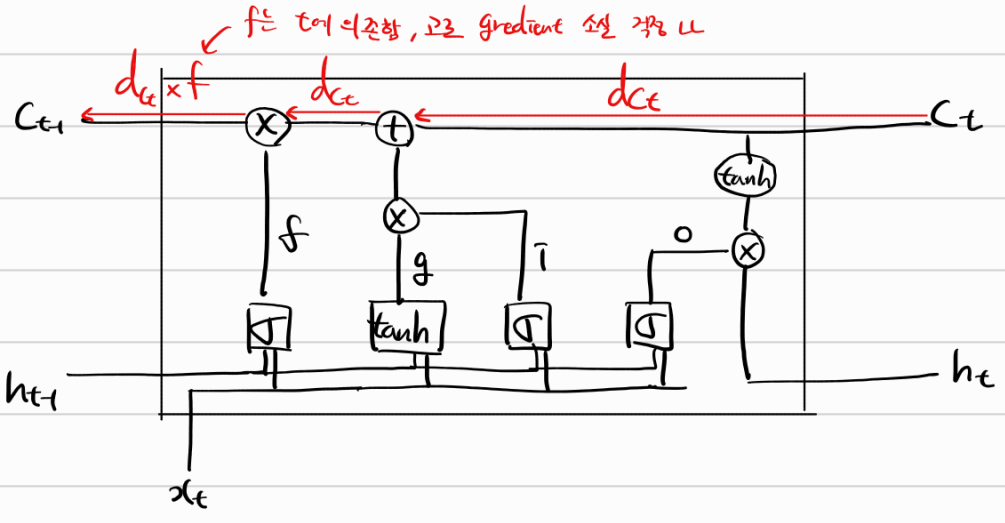
~~g~~ ← 기억 셀에 더해질 새로운 기억

Def 2.5 Input gate

Let $i = \sigma\left(x_t W_x^{(i)} + h_{t-1} W_h^{(i)} + b^{(i)}\right)$

Then $C_t \mathrel{+}= i \odot g$

Rmk 2.6 Memory cell's BP on LSTM



CH. 3 LSTM의 구현과 개선

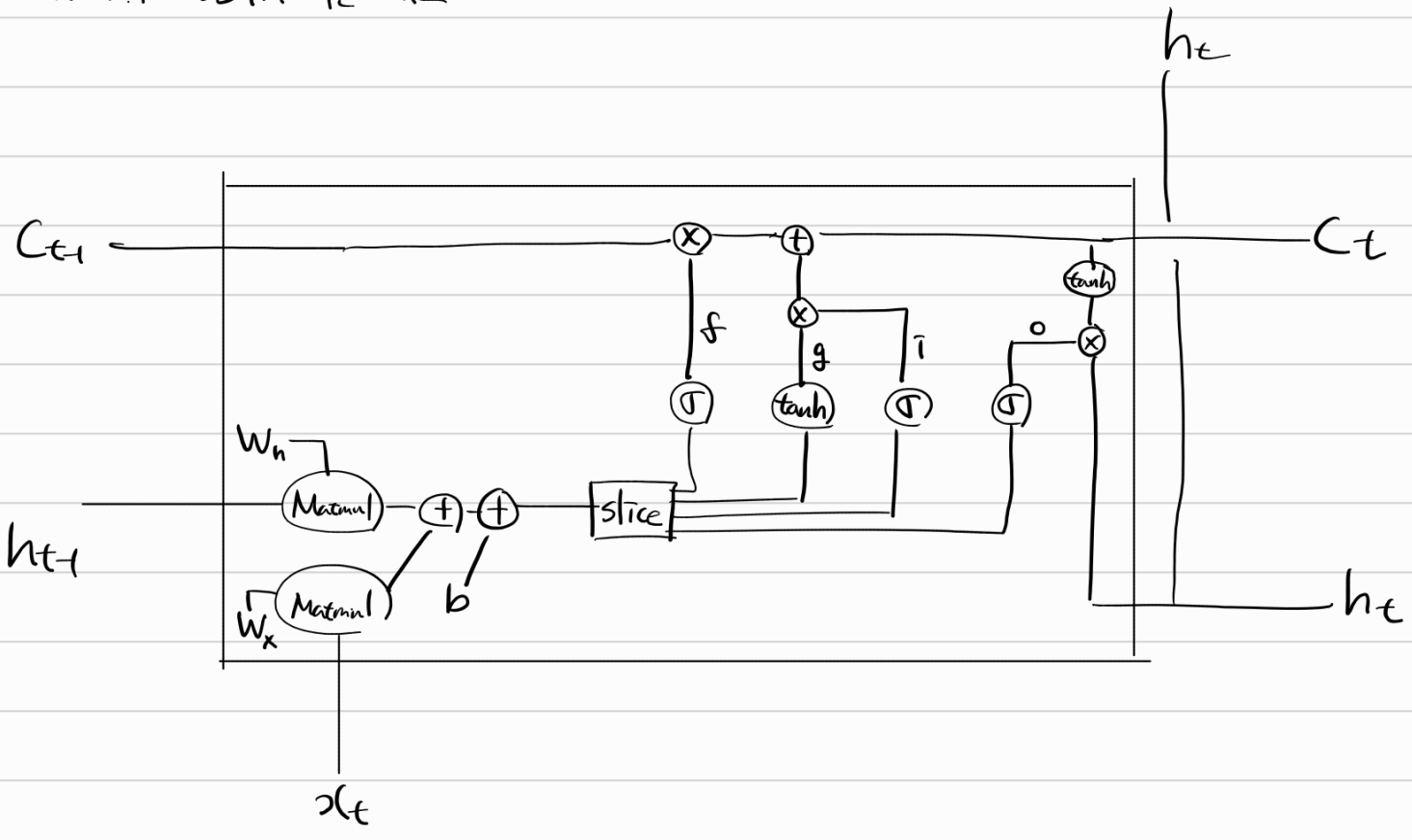
$$f = \sigma(h_{t-1}W_h^{(f)} + x_t W_x^{(f)} + b^{(f)})$$
$$g = \tanh(h_{t-1}W_h^{(g)} + x_t W_x^{(g)} + b^{(g)})$$
$$\bar{I} = \sigma(h_{t-1}W_h^{(i)} + x_t W_x^{(i)} + b^{(i)})$$
$$O = \sigma(h_{t-1}W_h^{(o)} + x_t W_x^{(o)} + b^{(o)})$$

$$C_t = f \odot C_{t-1} + \bar{I} \odot g$$

$$h_t = \tanh(C_t)$$

$$\Rightarrow X_t \underbrace{[W_x^{(f)} W_x^{(g)} W_x^{(i)} W_x^{(o)}]}_{W_x} + h_{t-1} \underbrace{[W_h^{(f)} W_h^{(g)} W_h^{(i)} W_h^{(o)}]}_{W_h} + \underbrace{[b^{(f)} b^{(g)} b^{(i)} b^{(o)}]}_{b}$$

Note 3.1 LSTM 계산 그래프

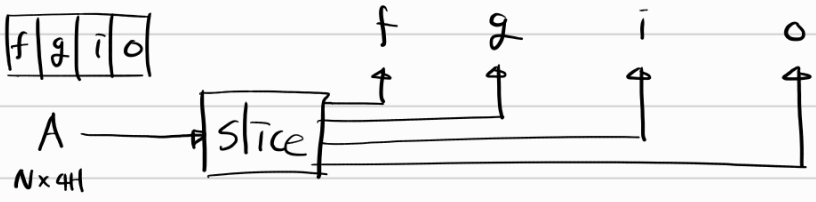


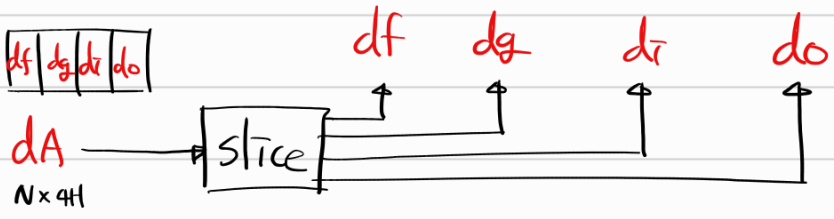
Thm 3.2 matrix size of LSTM

$$X_t W_x + h_{t-1} W_h + b$$

$N \times D \quad D \times 4H \quad N \times H \quad H \times 4H \quad N \times H$

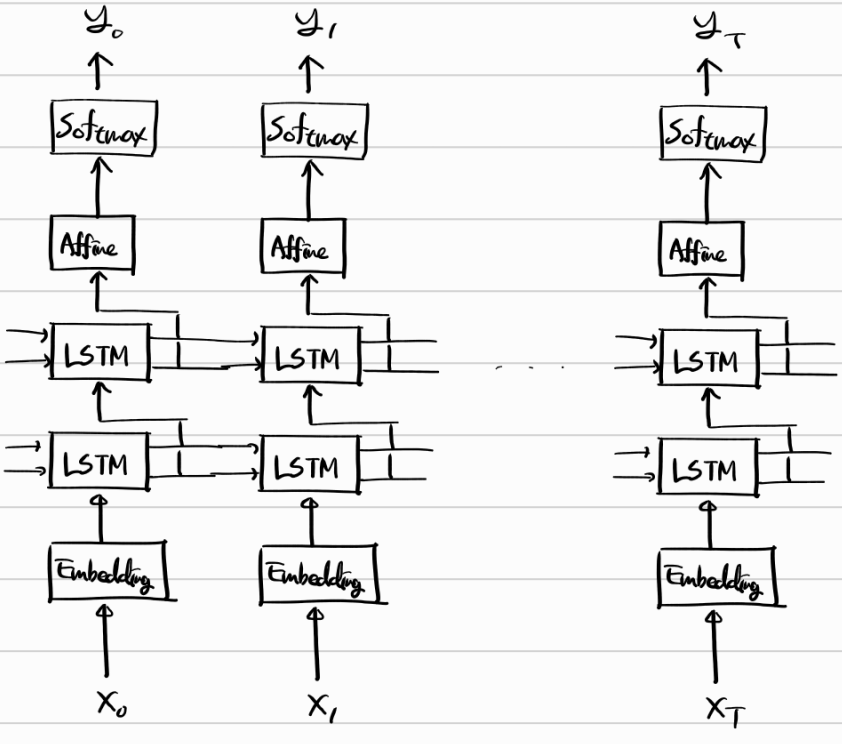
Thm 3.3 BP of LSTM slice





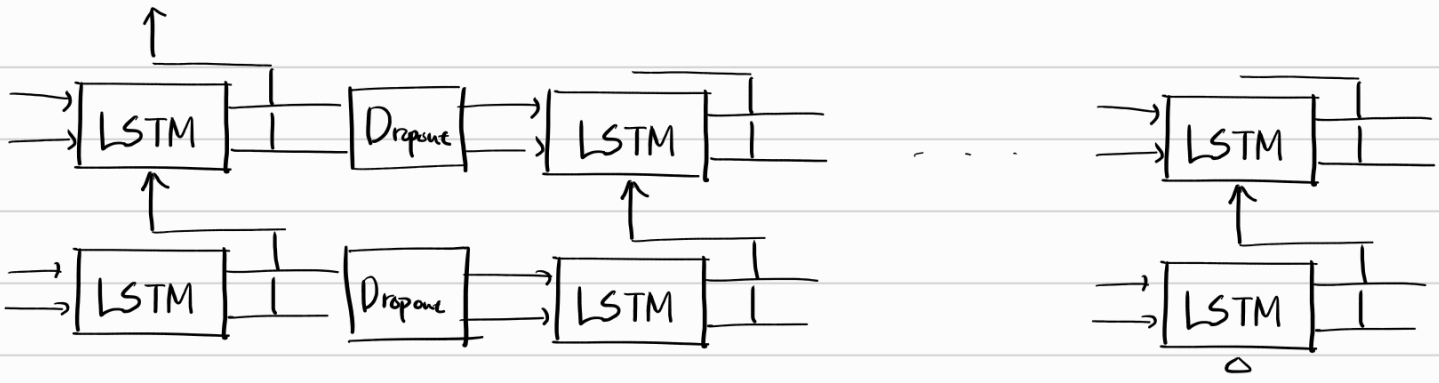
$\Rightarrow$ np.hstack(df, dg, di, do)

Def 3.4 LSTM based RNNLM

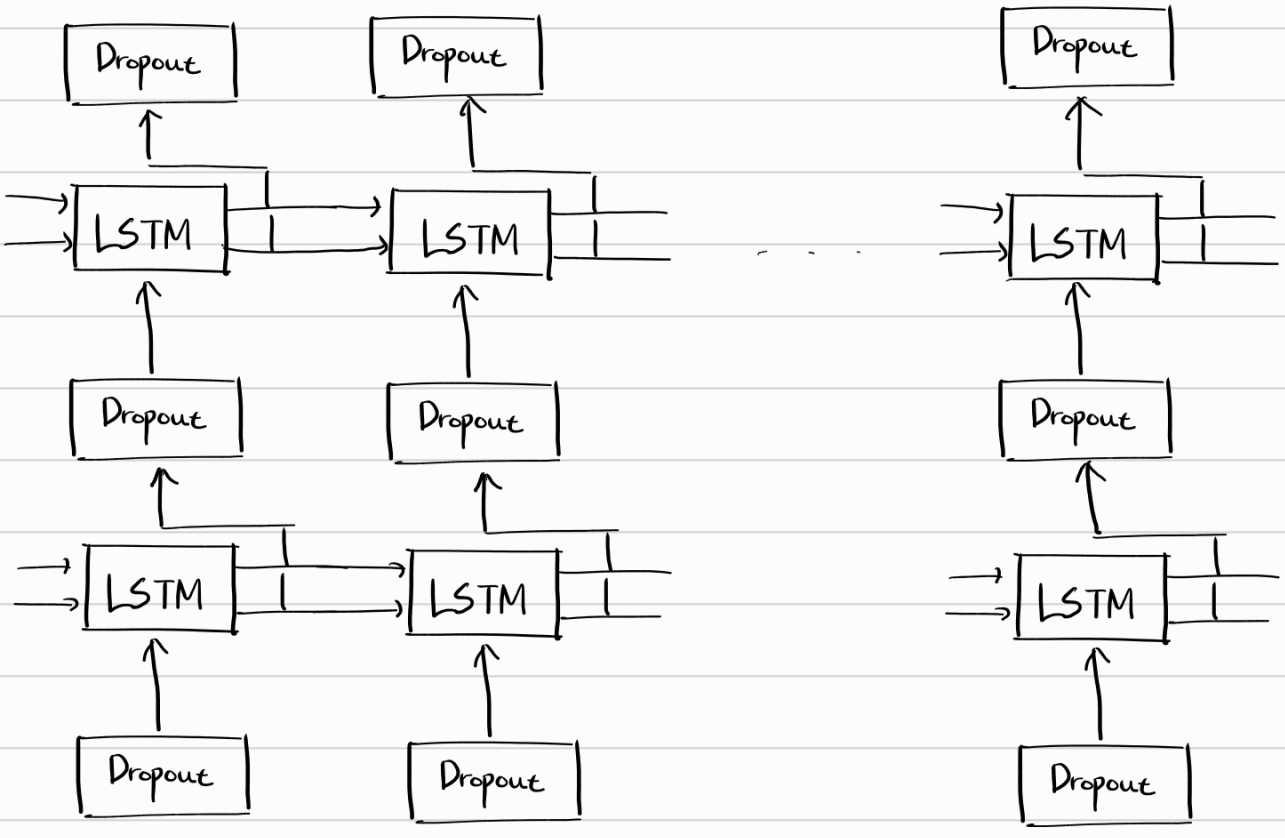


Thm 3.5 Dropout on LSTM

Option ① : 시계열 방향 삽입, 이는 학습에서 시간이 지남에 따라 정보가 사라질 수 있다. 즉, 흐르는 시간에 비례해서 노이즈 축적



option ② : 상하 방향 삽입



=) 시간에 따른 정보손실 ×

option ③ Variational Dropout

option ① + ② 에서 같은 포지션 레이어끼리 마스크를 공유!

+ Drop Connect 라는 기술도 있음 ...

Thm 3.6 Weight tying

먼저 매개변수를 줄이면 좋은 것 ① 학습 시간 ↓

② Overfitting 방지