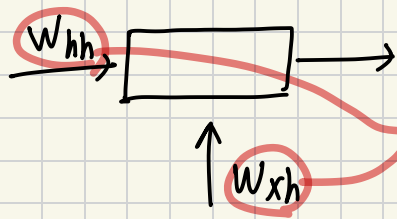


RNN 장점 : Input sequence 길이에 영향 X

단점 : 부정확치 X, vanishing gradient, long-term dependence



매 step Same 가중치를 사용하기에 입력길이 무관하게 계산이 가능하랴.

$$h_{t+1} = \tanh(W_{xh} \cdot x_{t+1} + W_{hh} \cdot h_t)$$

Backpropagation

$$\frac{\partial h_{t+1}}{\partial h_t} \Rightarrow \tanh'(W_{xh} \cdot x_t + W_{hh} h_t) \cdot W_{hh}$$

→ 매 step  $W_{hh}$ 가 곱해지게 된다.

0 ~ 1

축소됨.

모든 step에 대해서 (일반화)

$$\frac{\partial L}{\partial h_t} = \left( \prod_{k=2}^t \tanh'(W_{hh} h_{k-1} + W_{xh} x_k) \right) \cdot W_{hh}^{t-1} \cdot \frac{\partial h}{\partial h_w}$$

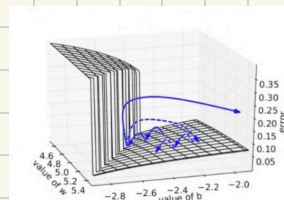
(0~1)을 계속 곱해서 0으로 수렴

$W_{hh} > 0$  이면 폭발  
 $W_{hh} < 0$  이면 소실

점진적 gradient가 폭발하거나, 소실될수 밖에 없다. → 특히 Layer가 깊어질수록

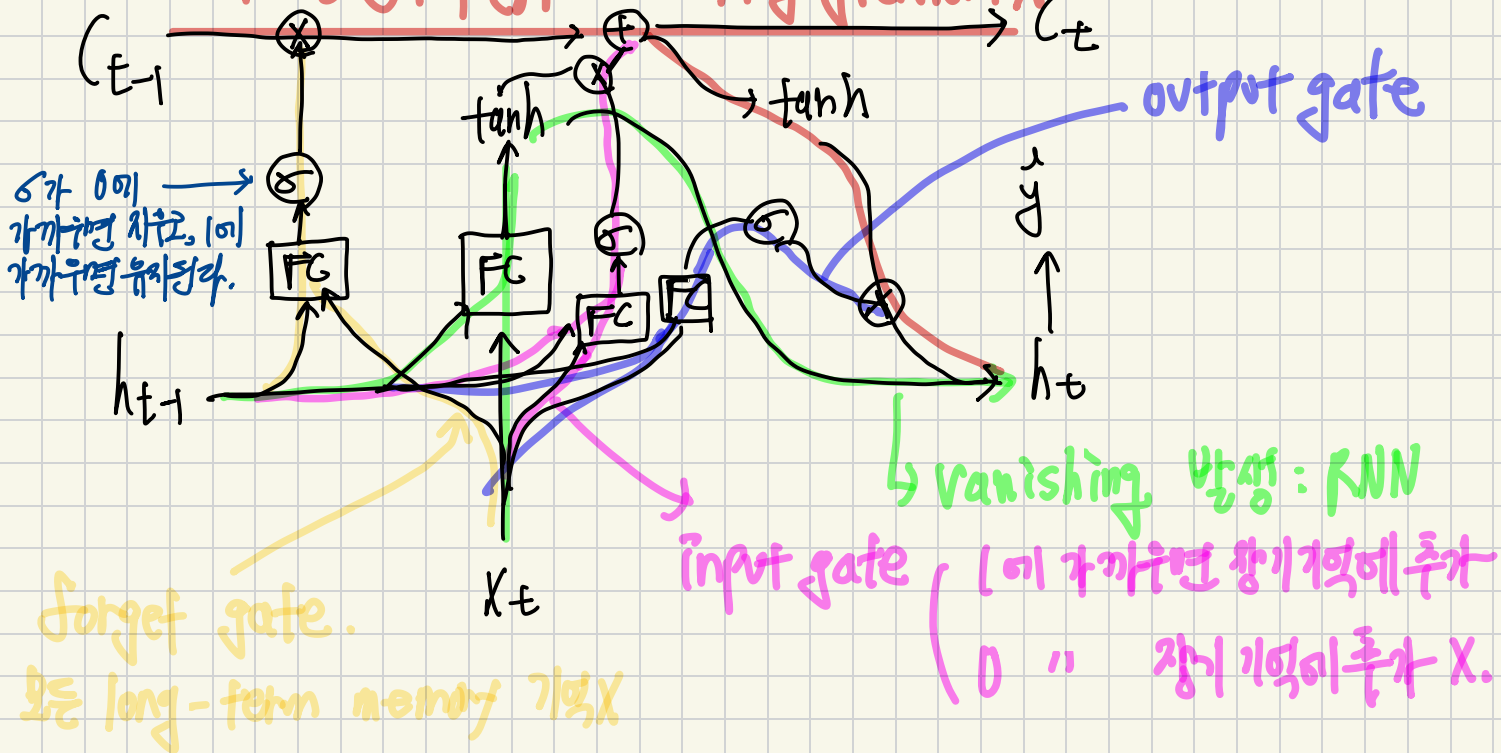
\* CNN은 Layer 때문에 가중치 분배 발생 But RNN에서는 입력길이 때문

① explode 방지 : clipping → gradient가 특정 크기를 넘기면 크기를 줄여준다.  
< 방법설명만 위지 >

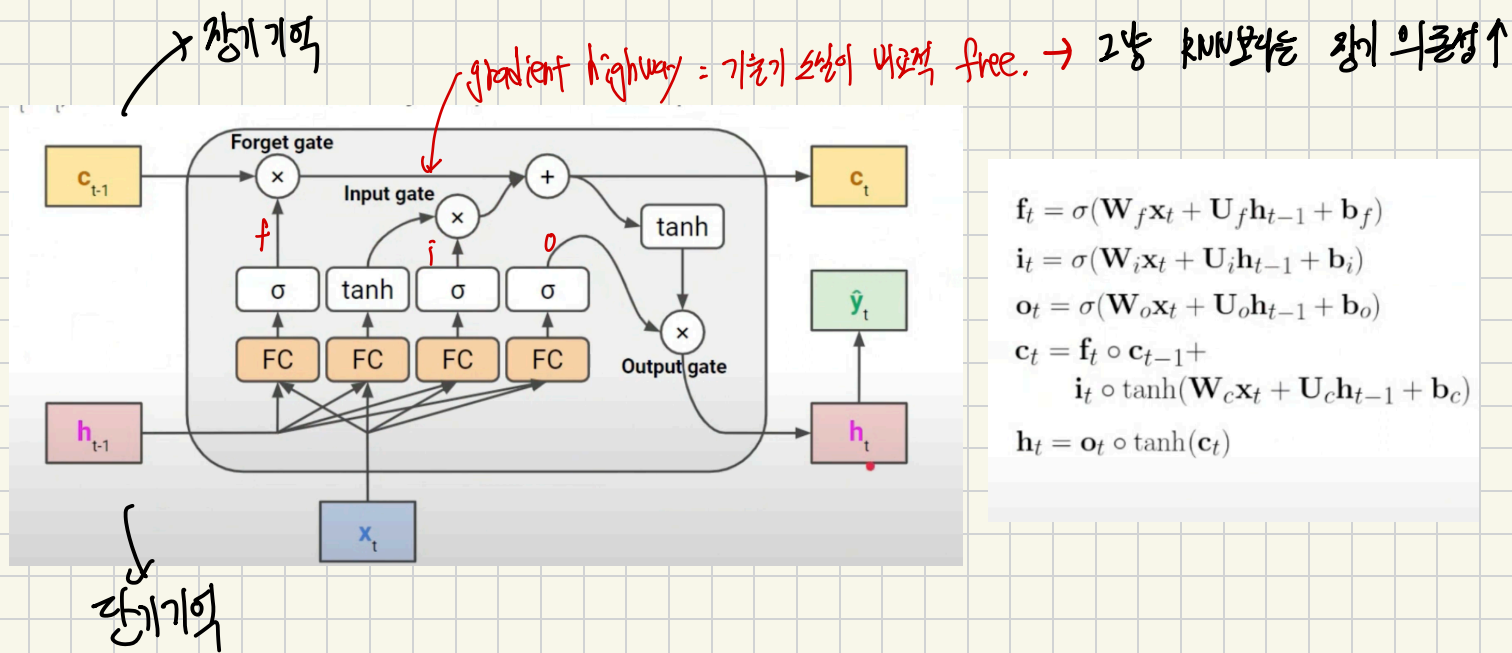


# <LSTM>

FC를 만지지 않고 vanishing gradient X

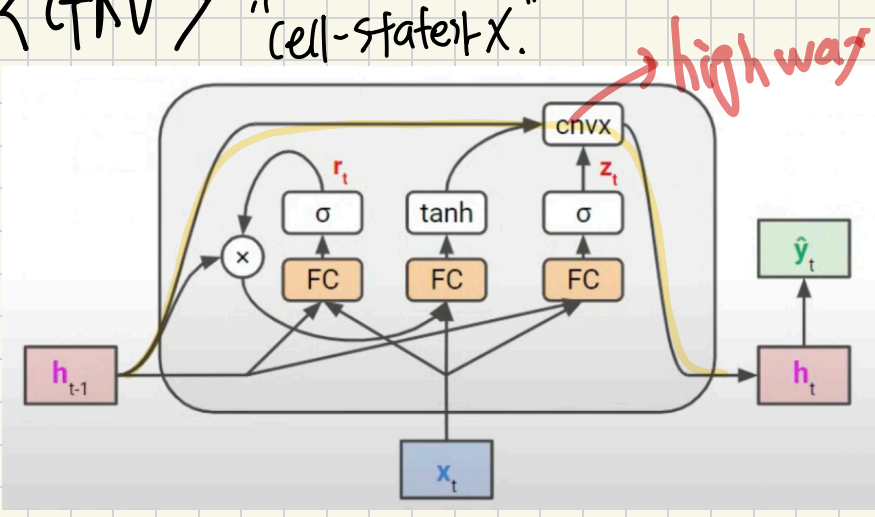


! 총 4개의 FC가 추가되었다.



But 완전히 vanishing, explode 문제를 해결 X, 어느정도 control.

# < GRV > "cell-state" X.



$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)$$

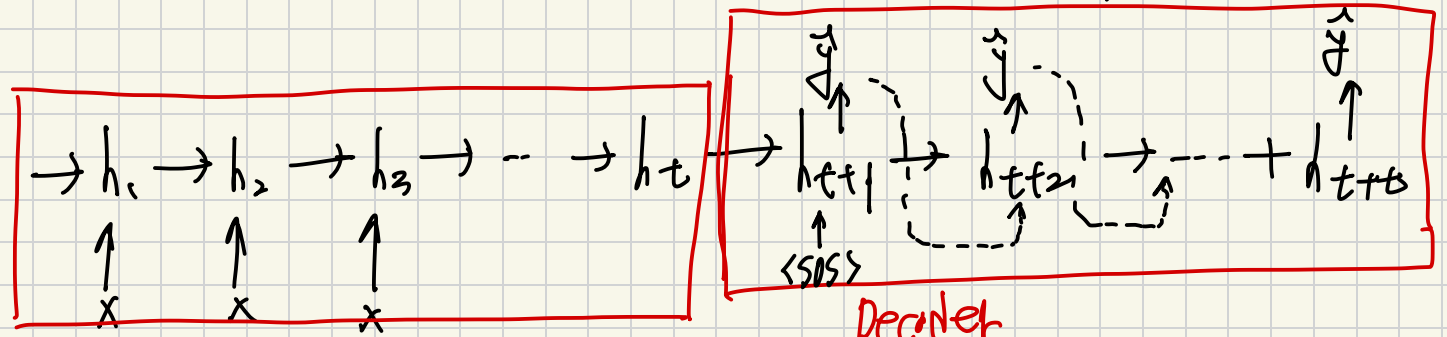
$$h_t = (1 - z_t) \circ h_{t-1} + z_t \circ \tanh(W_h x_t + U_h (r_t \circ h_{t-1}) + b_h)$$

Convex combination. 이전 정보와 현재 정보를 균형을 맞추기 위해.

LSTM과 비슷하지만 GRU의 parameter 수가 적어서 학습이 빠르다.

# < Seq2Seq > Input & output 1:1 matching 안되는 문제 해결.

가계번역시 Input과 output의 단어가 다르기 때문에 1개의 입력에 1개의 출력이 매칭X. 즉, 1대1 매칭이 안된다. => 어느정도 sentence 정보를 갖고, 출력을 내보내라.



Encoder  
문장을 읽는 역할.

Decoder.  
Auto Regressive 하기 (전 단계 output  
다음 단계 input)

Training에는 잘 안됨  
\*강제 Teacher forcing : 다음 input에 정답을 넣어준다.

