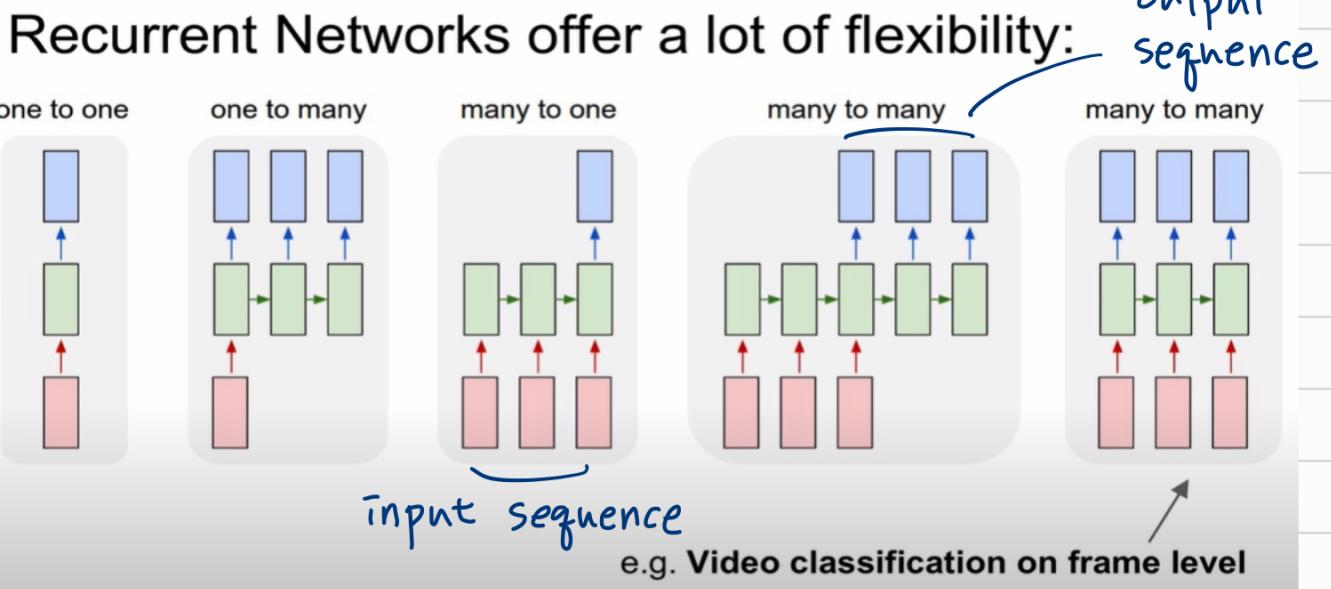


# RNN : Recurrent Neural Network



예측 : 현재 frame + 지난간 모든 frame 들에  
대한 함수!

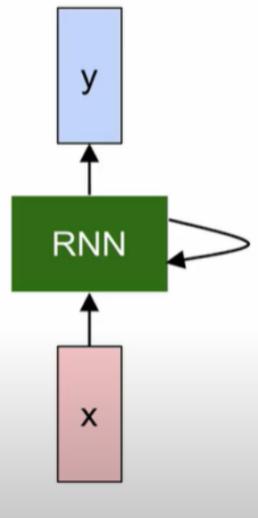
We can process a sequence of vectors  $\mathbf{x}$  by applying a recurrence formula at every time step:

recurrence function

$$h_t = f_W(h_{t-1}, x_t)$$

new state      old state      input vector at some time step

some function with parameters  $W$



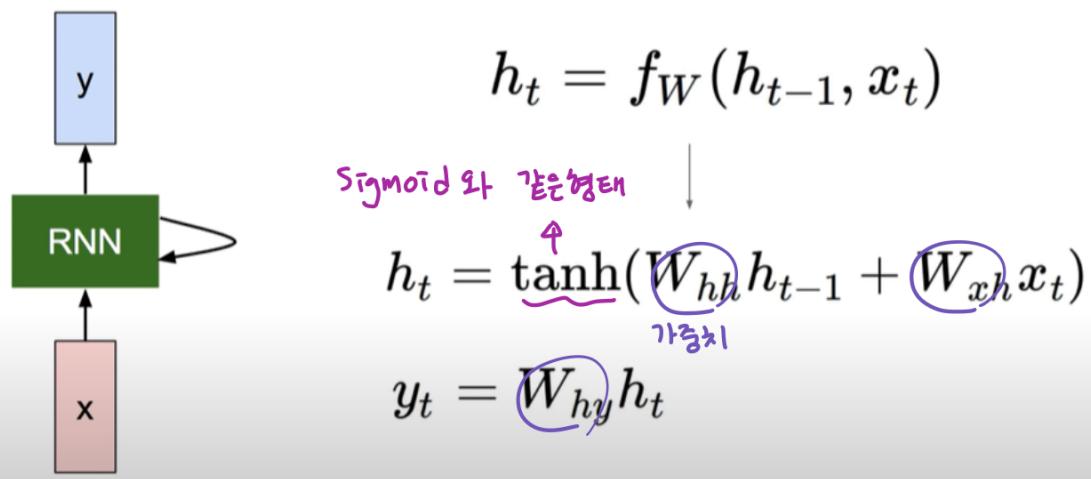
time stamp마다 동일한 function /  
parameter set 사용!

RNN : 이전 State + 현재 Input  $\rightarrow$  현재 State

입력                          출력

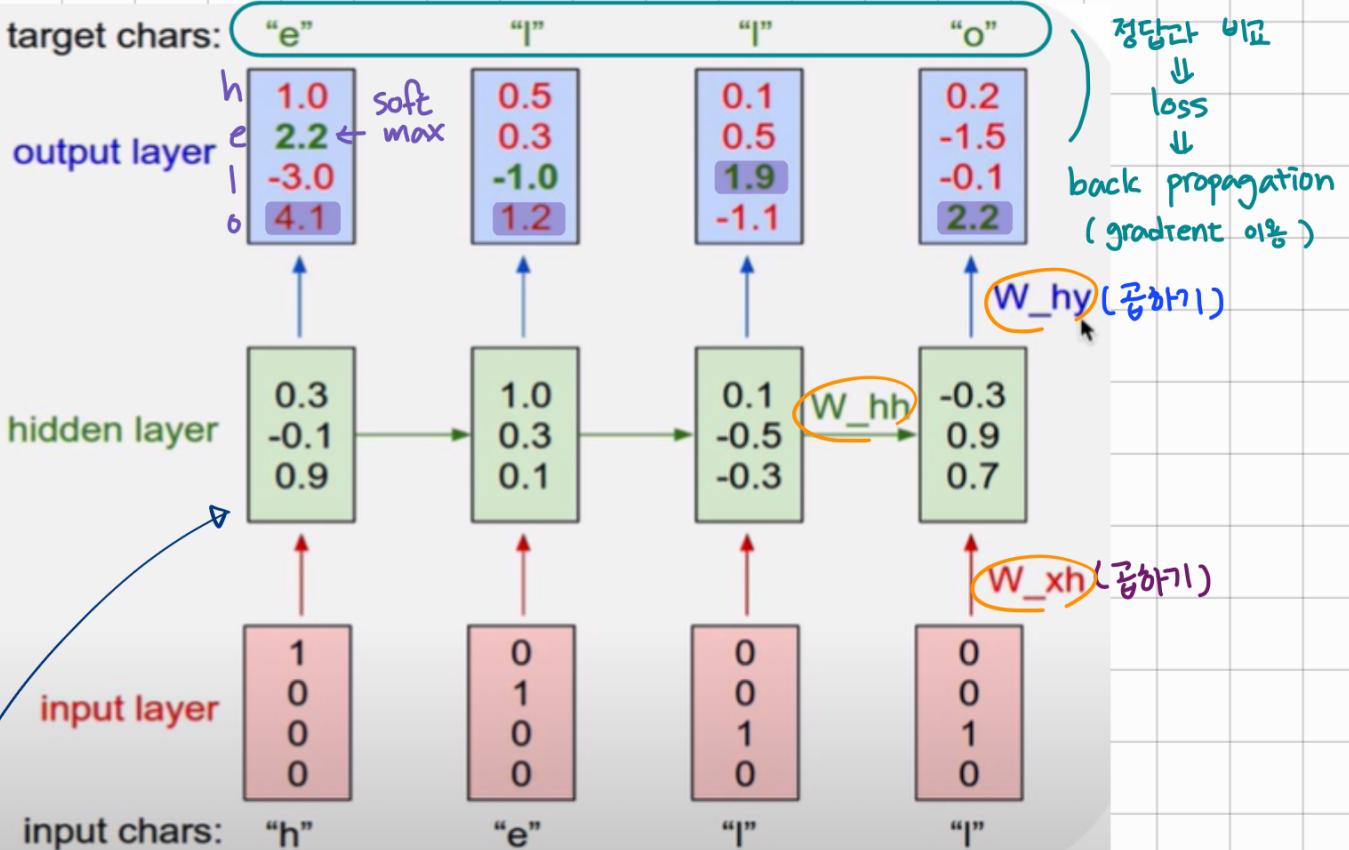
# (Vanilla) Recurrent Neural Network

The state consists of a single "hidden" vector  $\mathbf{h}$ :



\* "hello" 학습 → 다음 글자 예측해보자

$$\begin{matrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{matrix} \quad \begin{matrix} h \\ e \\ l \\ l \\ o \end{matrix}$$



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

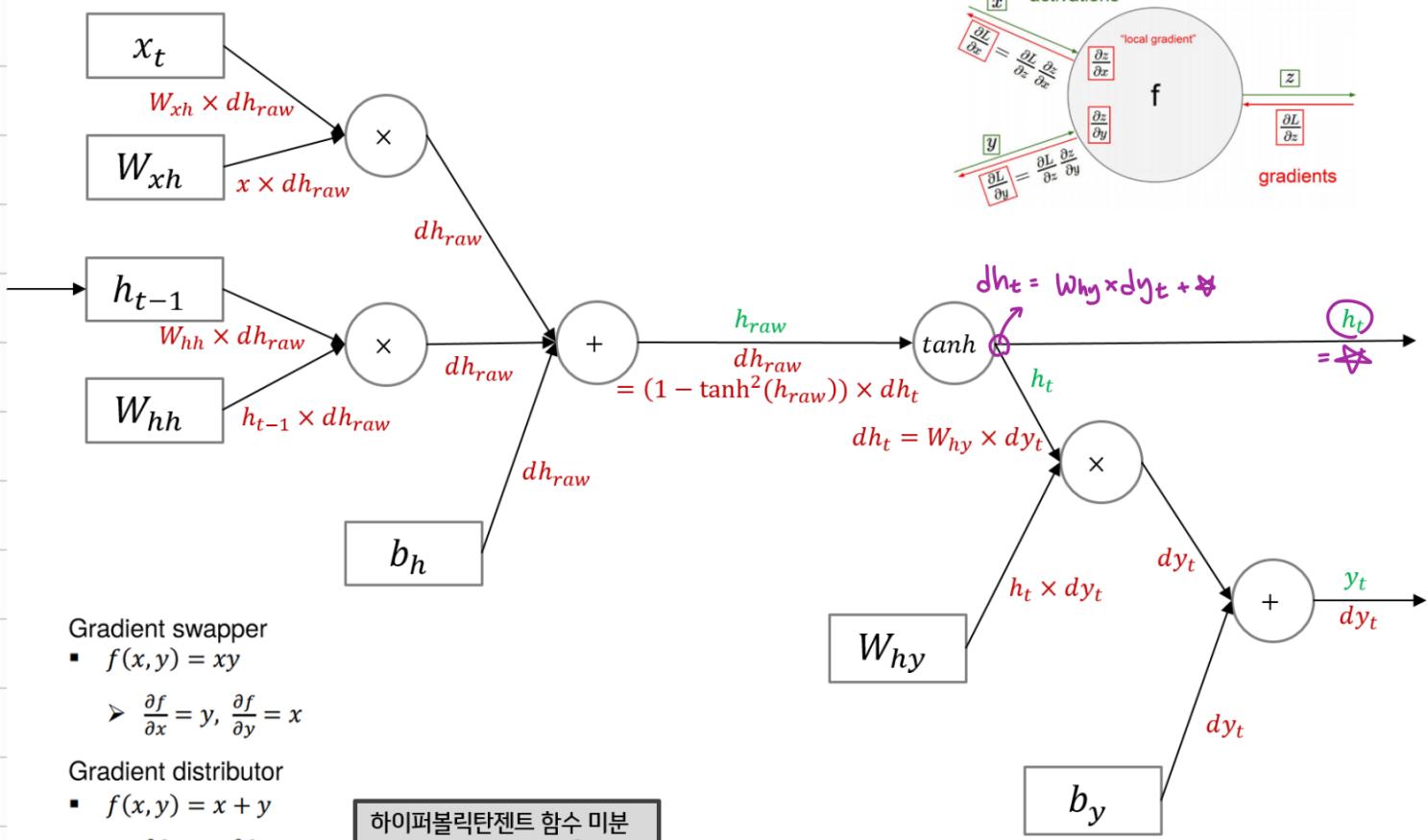
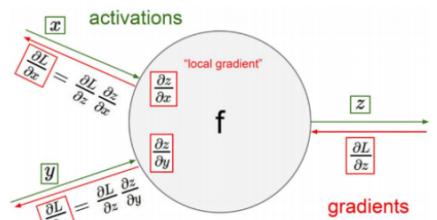
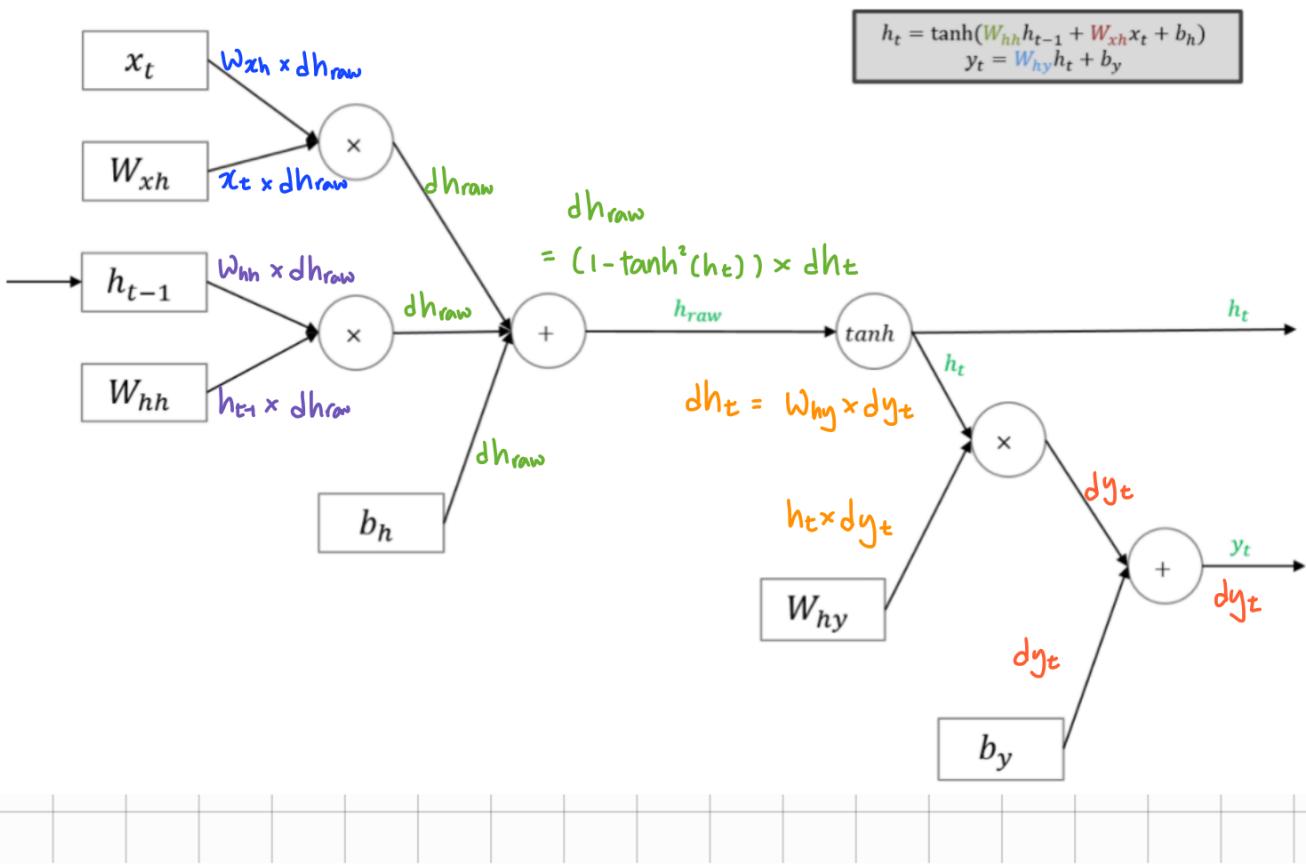
$\downarrow$   
업데이트

\*  $W_{hh}/W_{xh}/W_{hy}$   
는 t에 관계없이 일정!

$$y_t = W_{hy}h_t$$

$4 \times 3 \quad 3 \times 1$

# (RNN) 역전파



Gradient swapper

- $f(x, y) = xy$
- $\frac{\partial f}{\partial x} = y, \frac{\partial f}{\partial y} = x$

Gradient distributor

- $f(x, y) = x + y$
- $\frac{\partial f}{\partial x} = 1, \frac{\partial f}{\partial y} = 1$

하이퍼볼릭탄젠트 함수 미분  
 $\tanh(x) \rightarrow 1 - \tanh^2(x)$