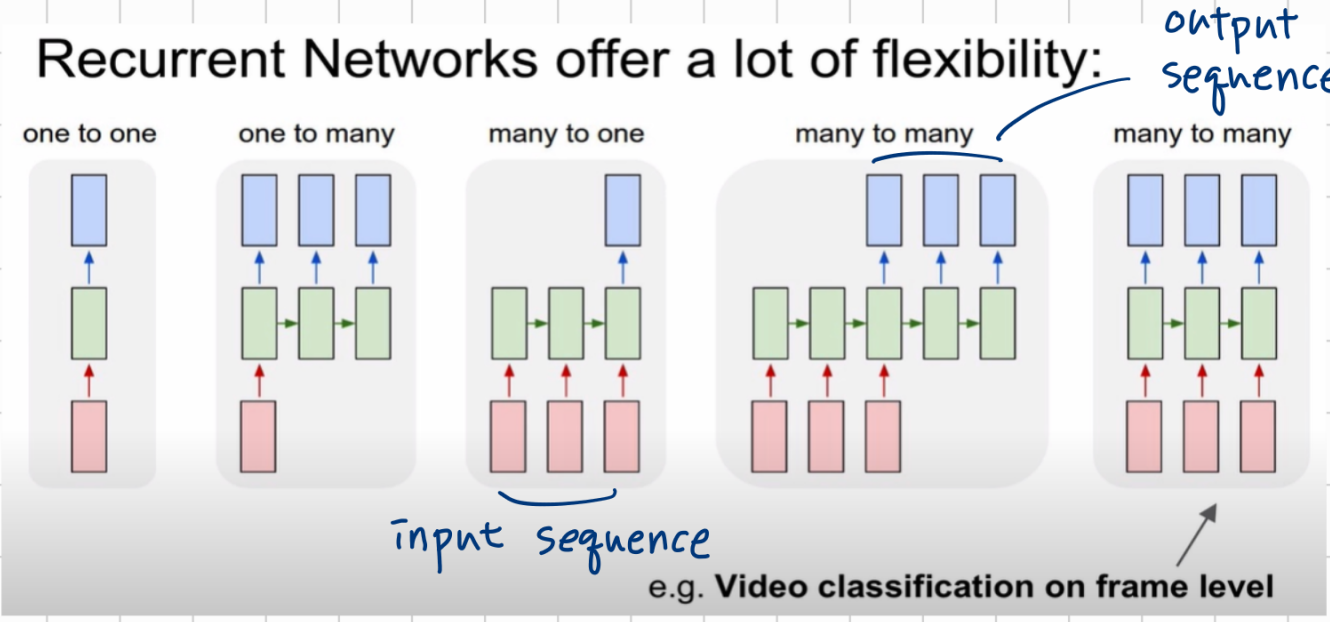
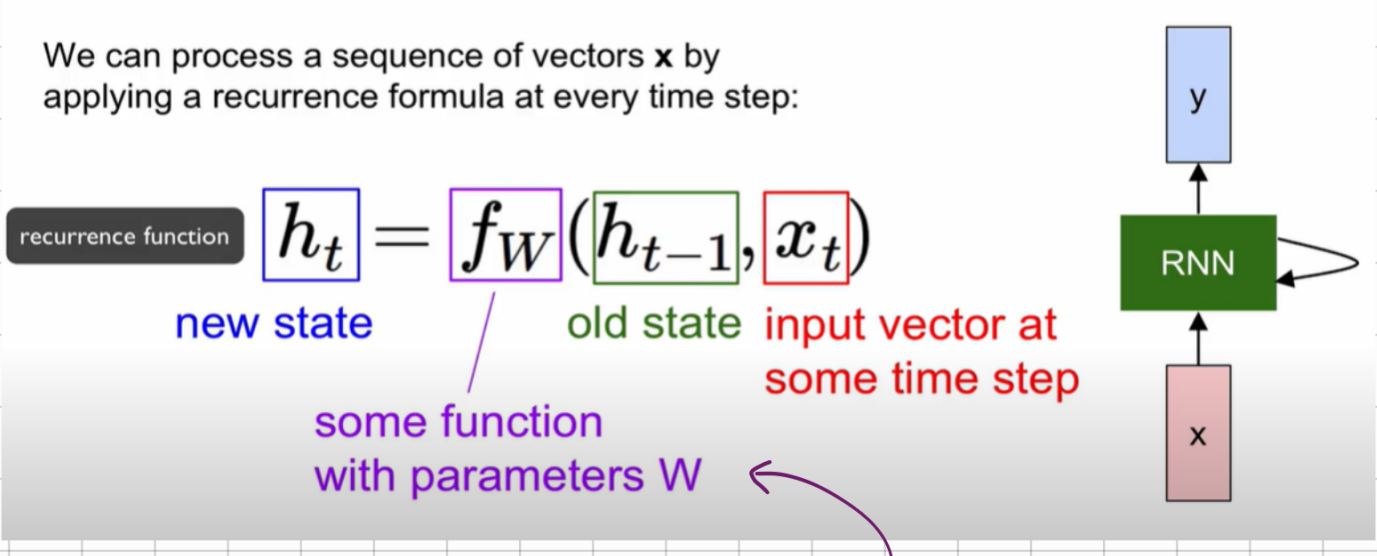
RNN : Recurrent Neural Network



예측 : 현재 frame + 지나간 모든 frame들에 대한 함수!



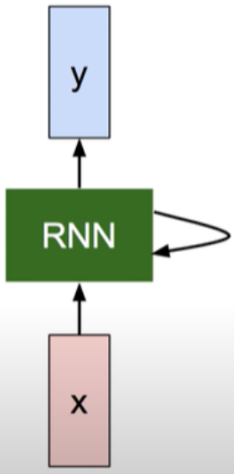
time stamp 마다 동일한 function / parameter set 사용!

RNN : 이전 state + 현재 input → 현재 state

입력 출력

(Vanilla) Recurrent Neural Network

The state consists of a single *"hidden"* vector **h**:



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

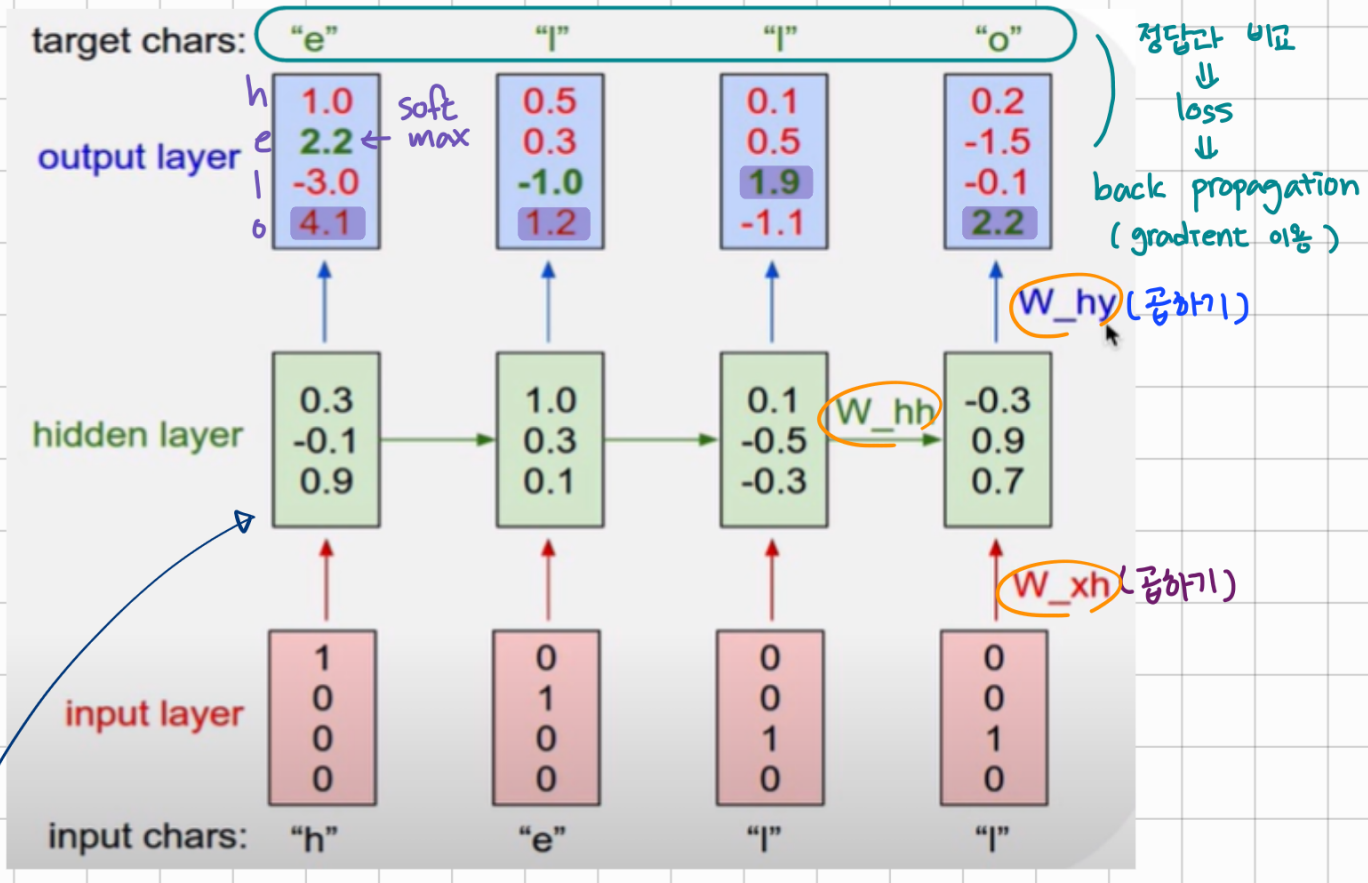
Sigmoid와 같은형태

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

가중치

$$y_t = W_{hy} h_t$$

* "hello" 학습 → 다음 글자 예측해보자



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

3×4 4×1

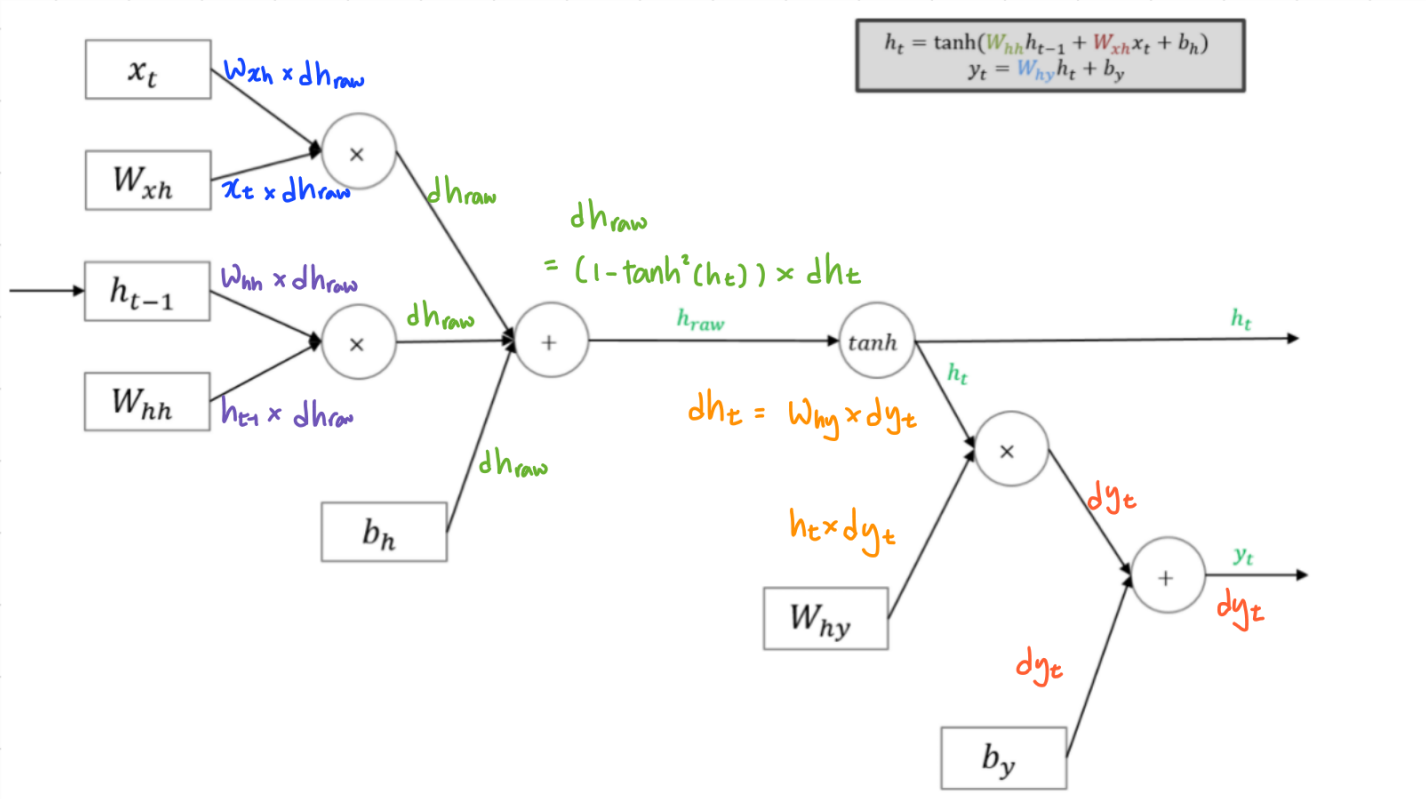
↓ 없으면 0

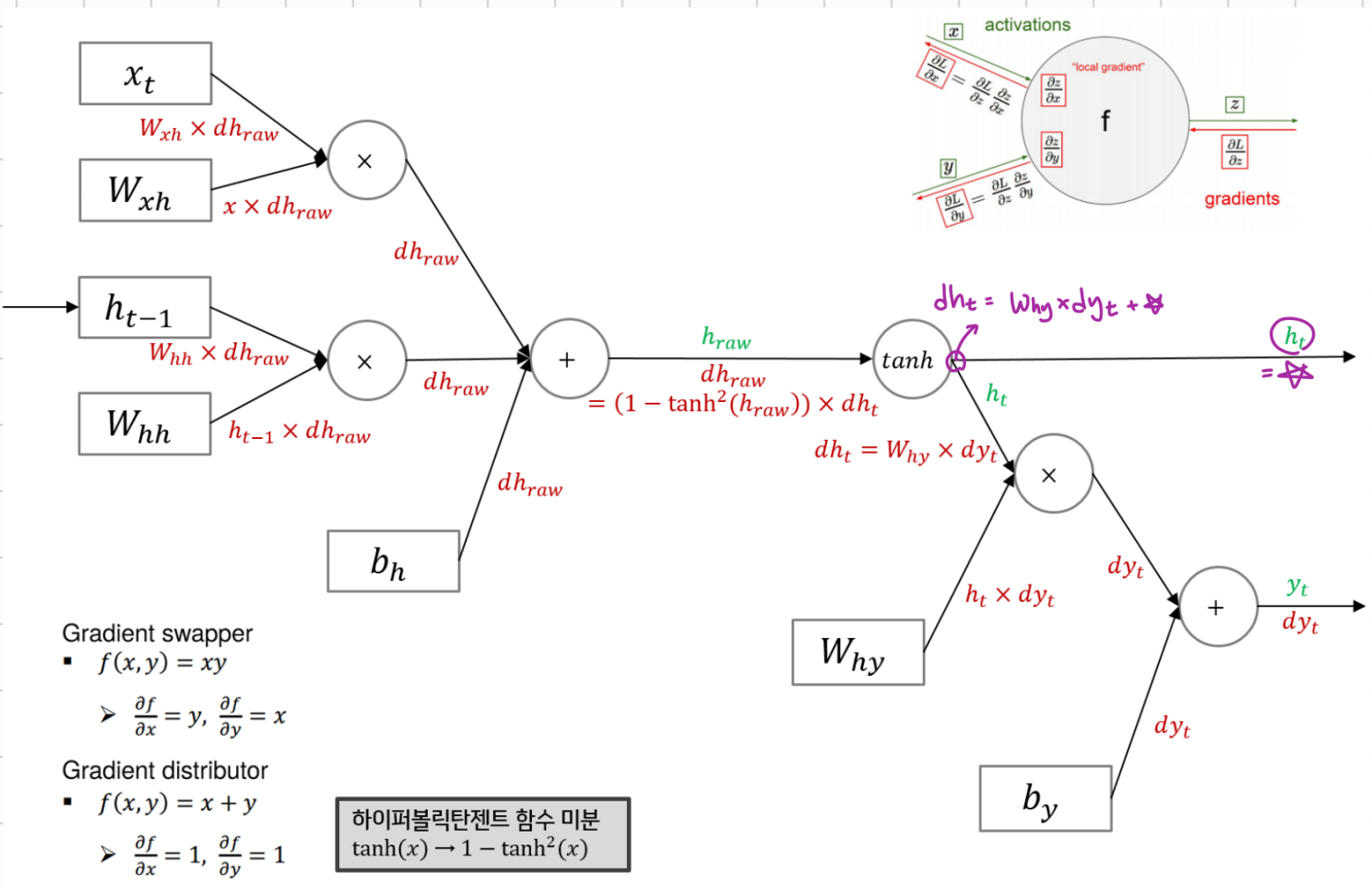
* W_{hh} / W_{hx} / W_{hy} 는 t에 관계없이 일정!

$$y_t = W_{hy} h_t$$

4×3 3×1

< 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
$\tanh(x) \rightarrow 1 - \tanh^2(x)$