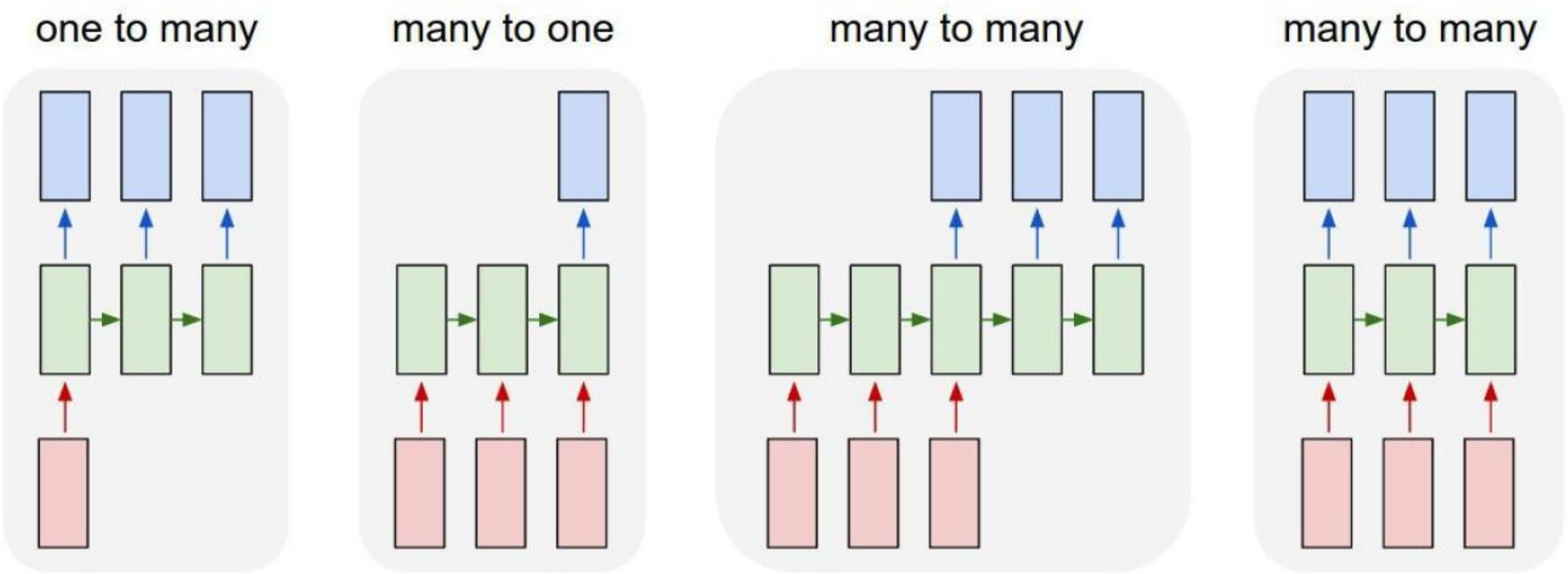
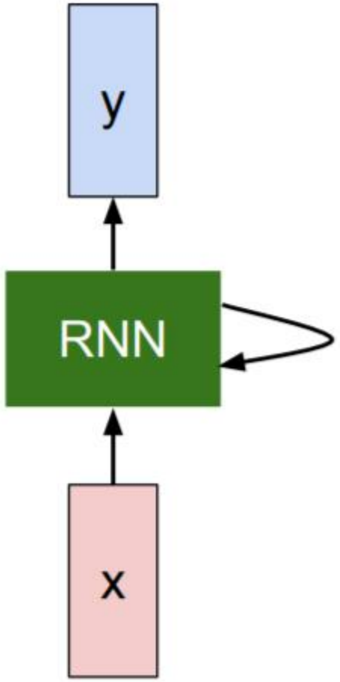
CS231N

LEC 10 | RECURRENT NEURAL NETWORK

- CNN 은 one to one 구조
- RNN 은 one to many, many to one, many to many 구조에 효과적.



VANILLA RECURRENT NEURAL NETWORK



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

$$\downarrow$$

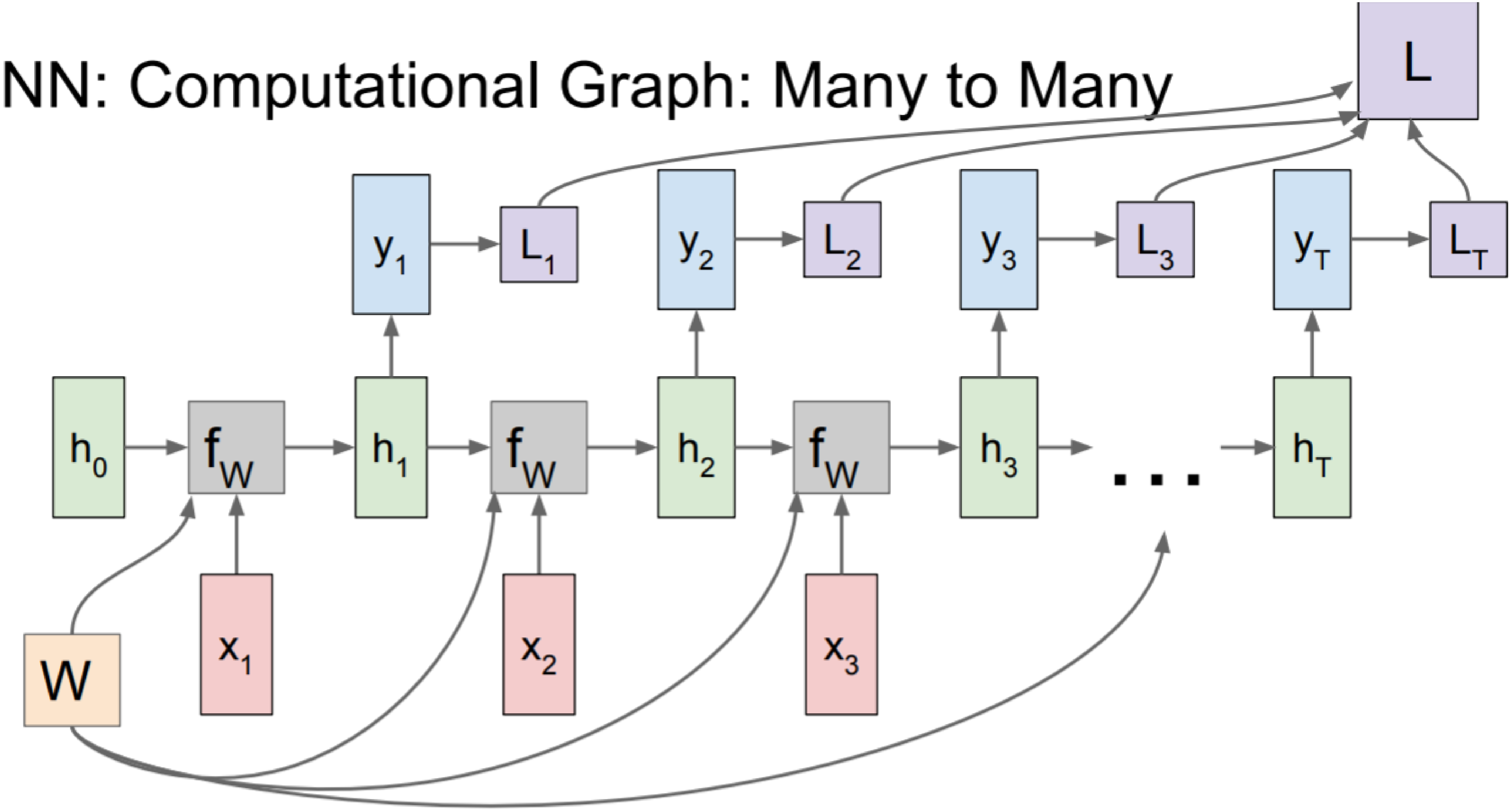
$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

$$y_t = W_{hy}h_t$$

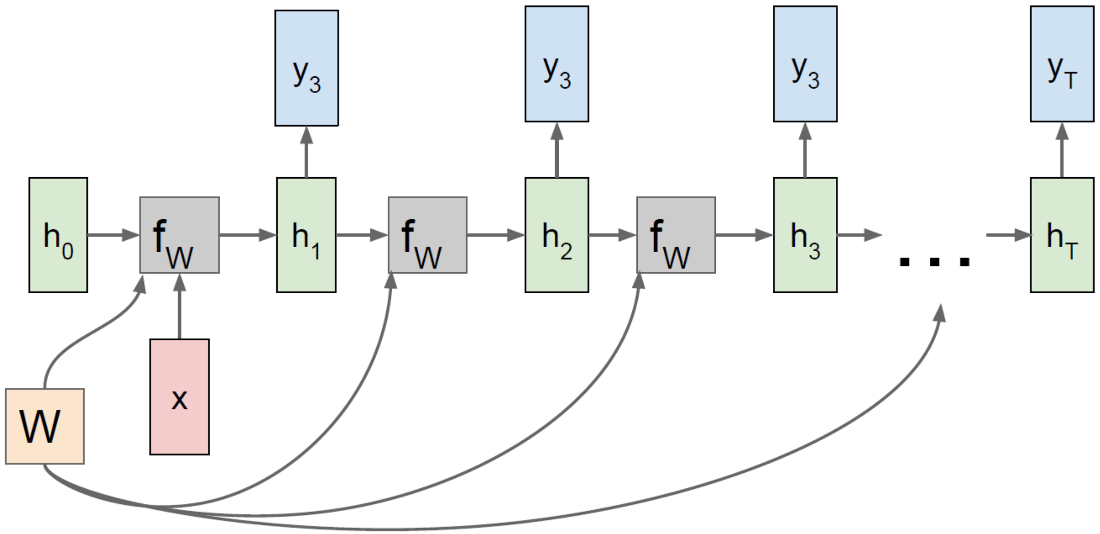
Why tanh?

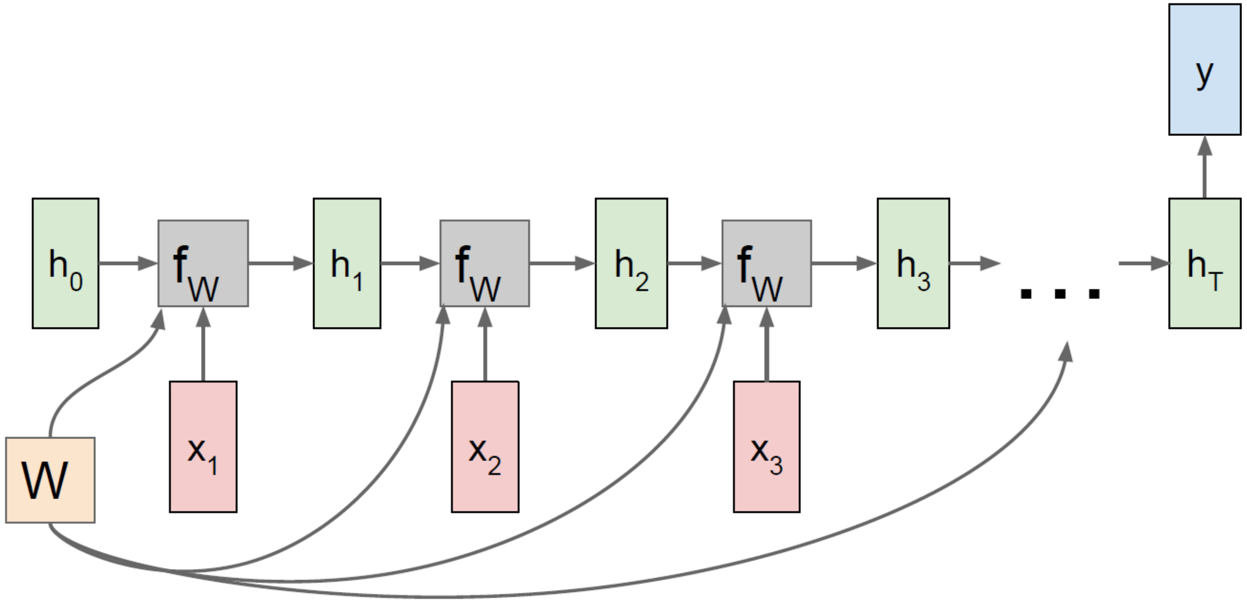
- Non-linearity.
- Zero-centered.
- No vanishing gradient.(architecture 특징)

RNN: Computational Graph: Many to Many

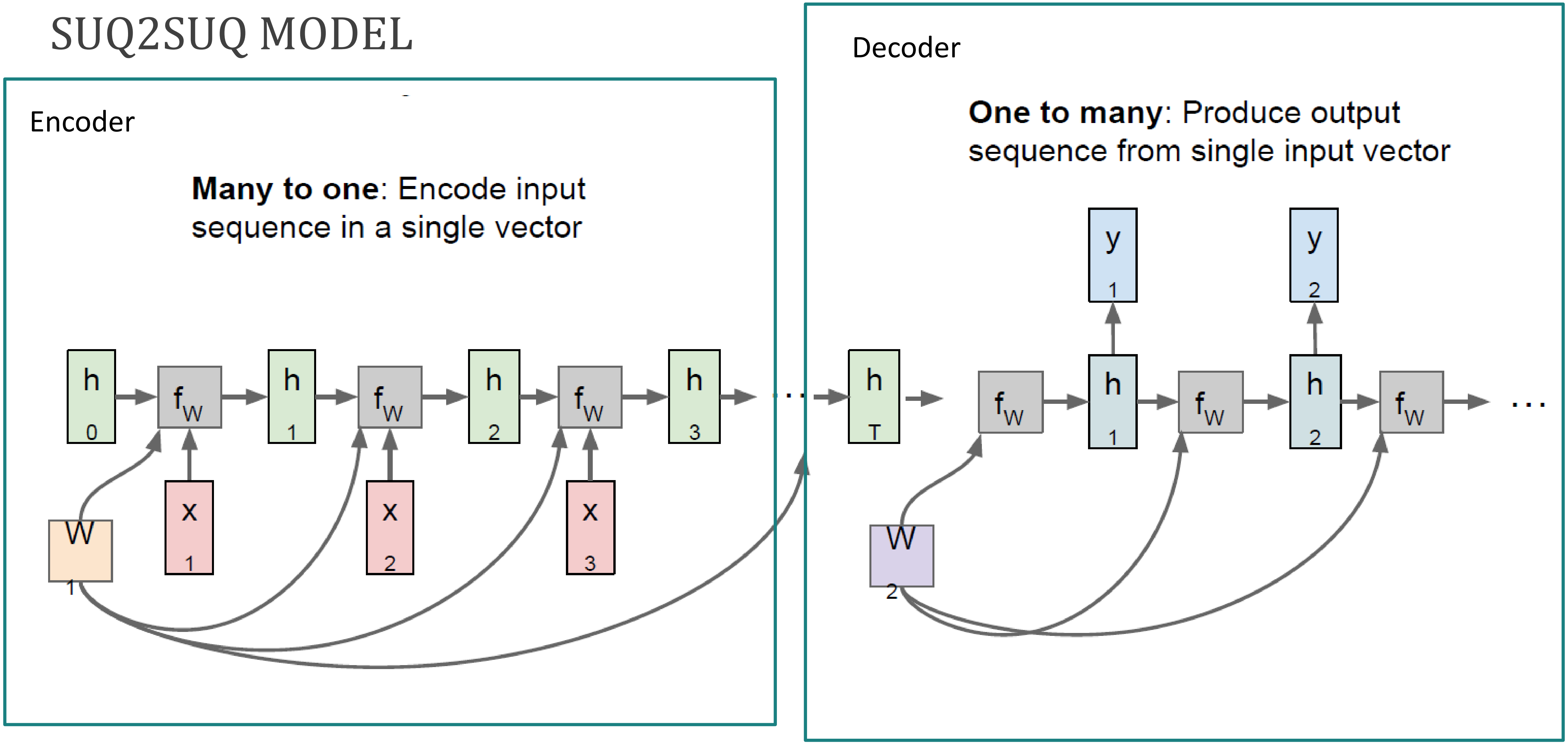


ONE TO MANY, MANY TO ONE



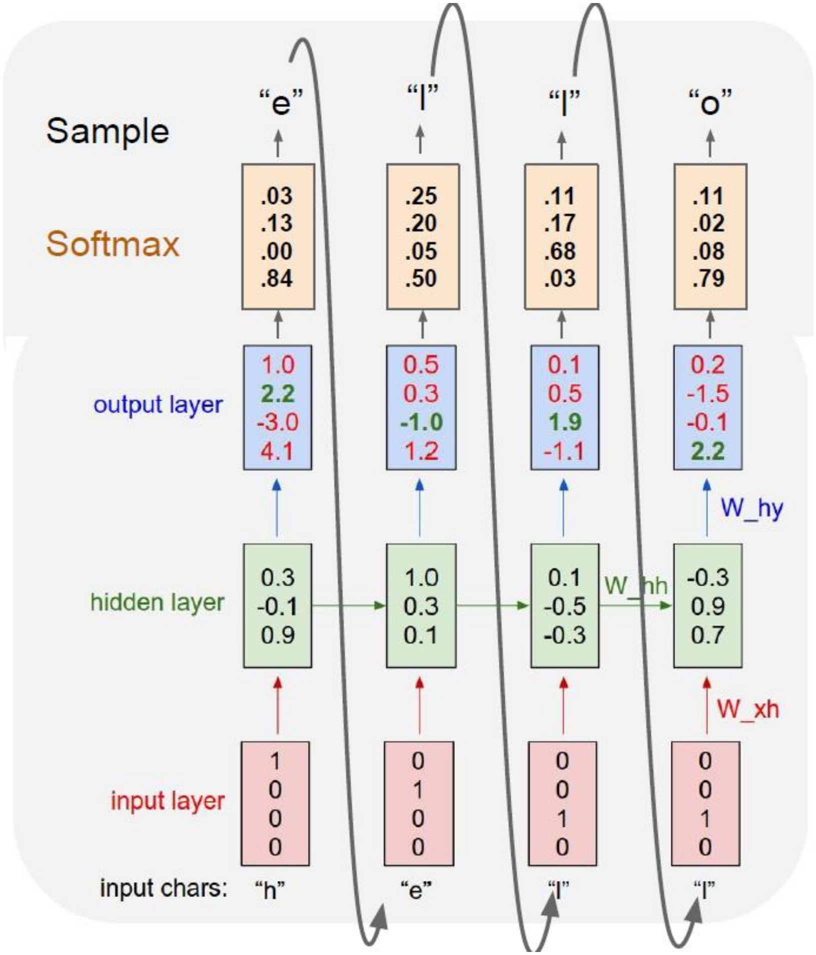


SUQ2SUQ MODEL



E.G) CHARACTER LEVEL LANGUAGE MODEL SAMPLING

Vocabulary:
[h,e,l,o]



BACKPROPAGATION THROUGH TIME

처음부터 끝까지 forward 하고 backpropa는 데이터 형태
에 따라 매우 비효율적일 수 있다. Mini batch 처럼!

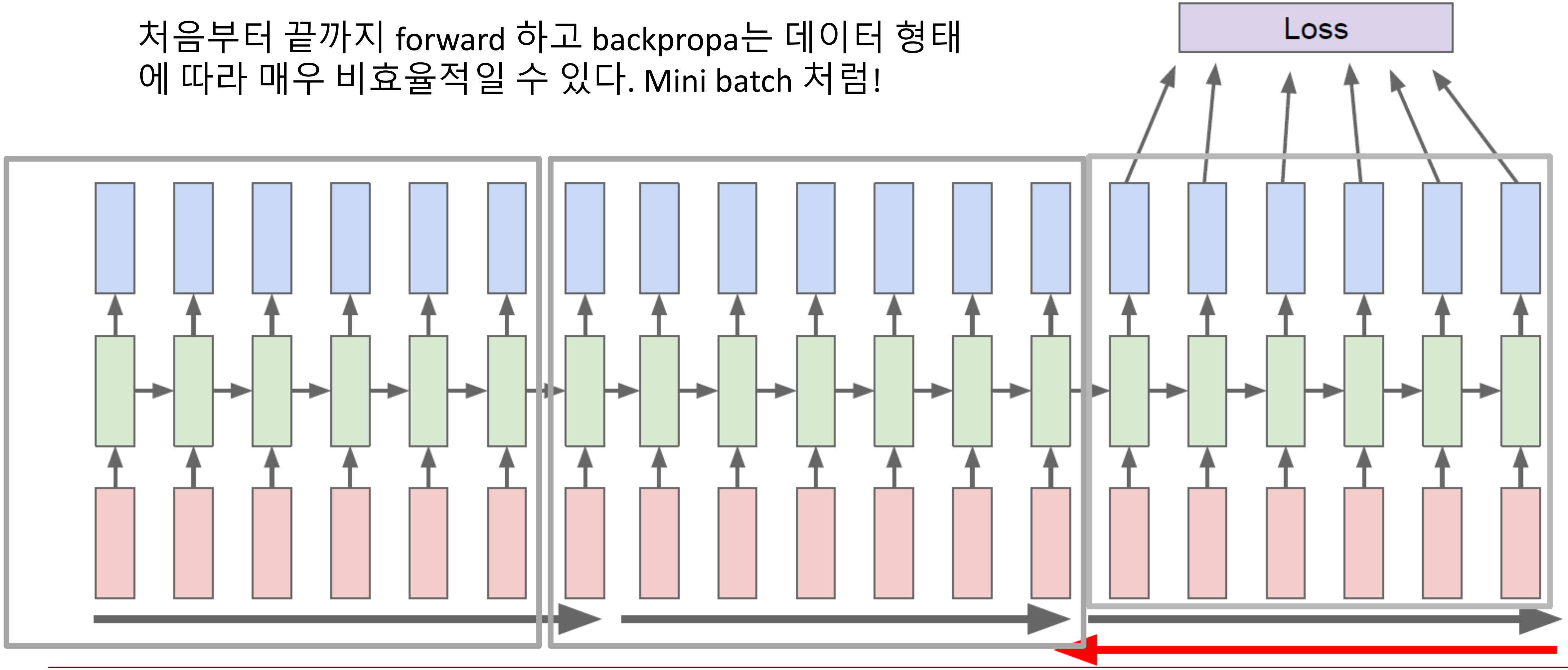
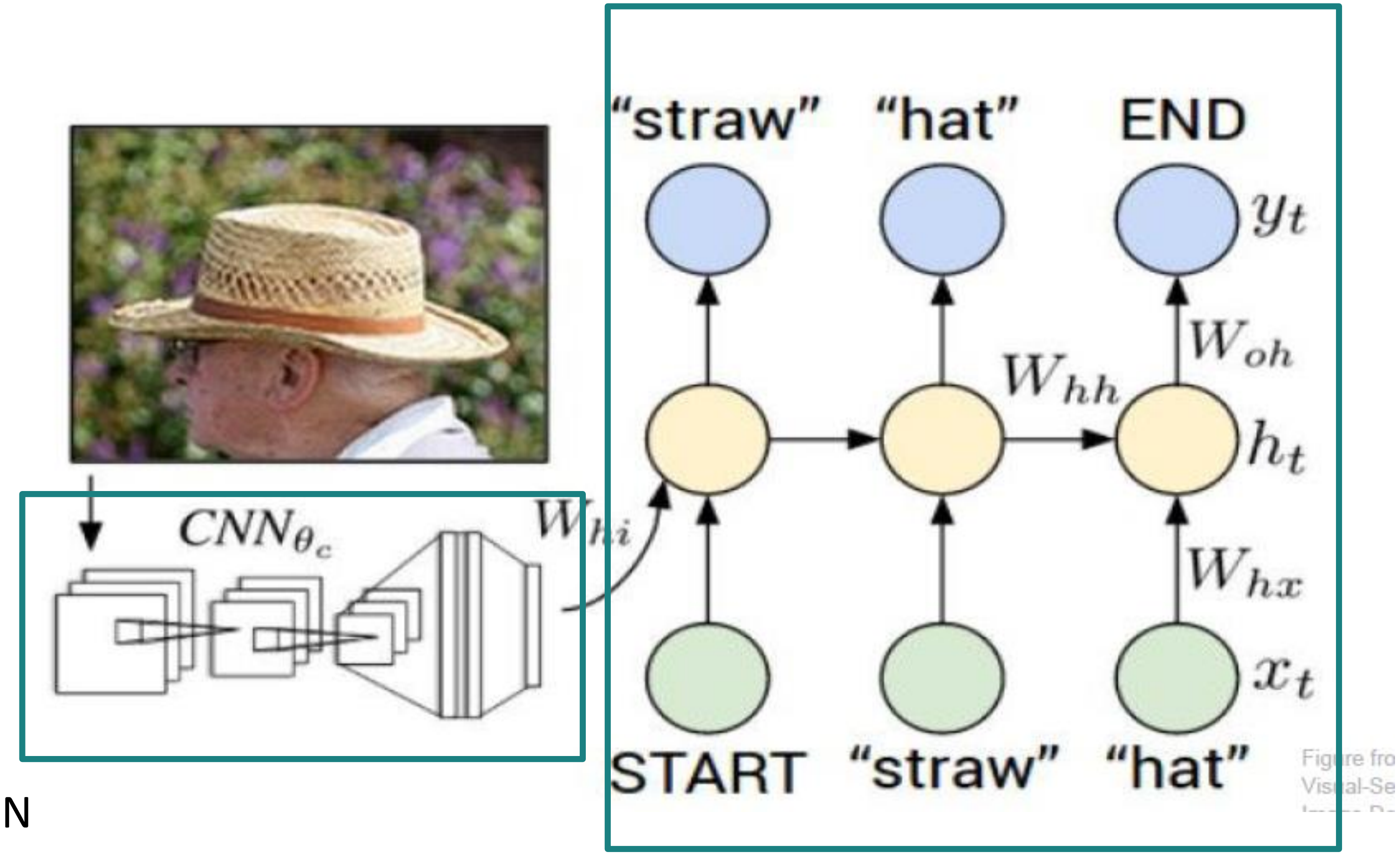
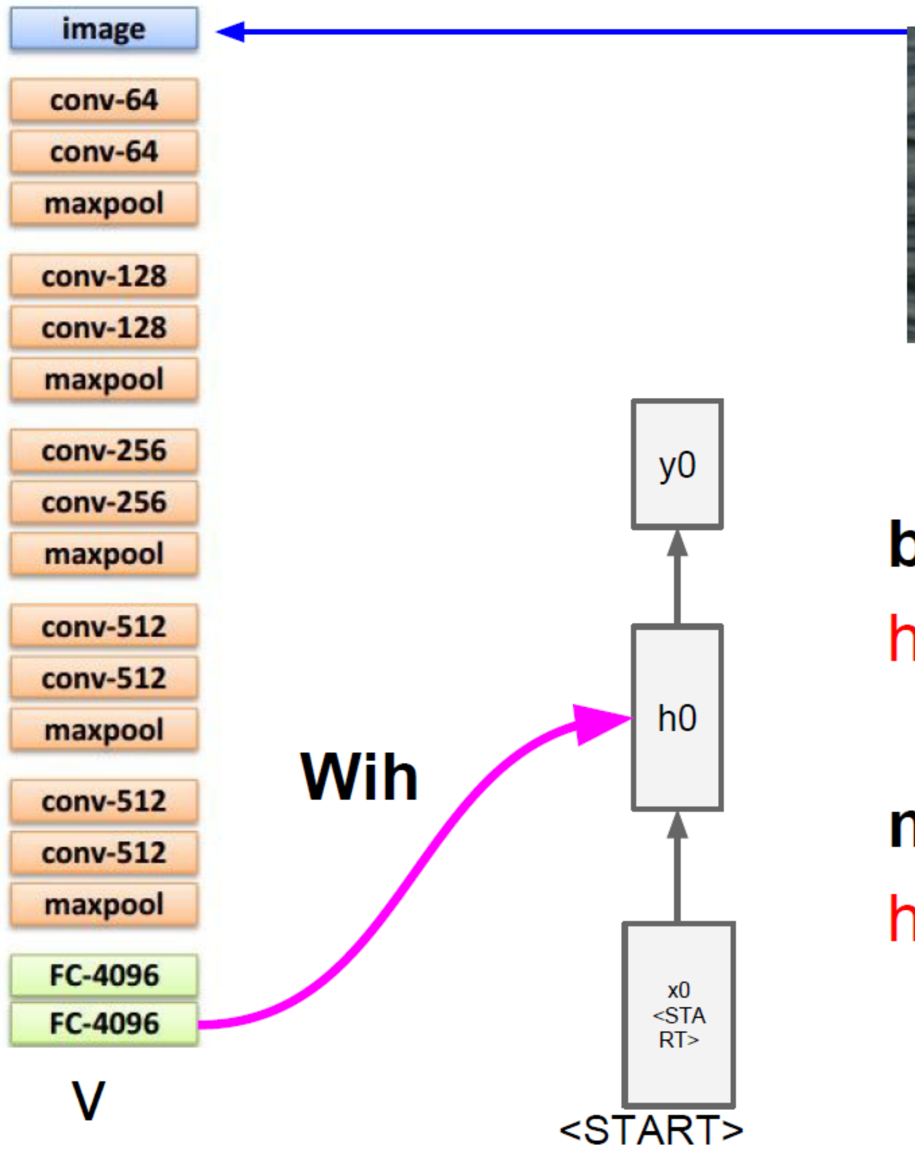


IMAGE CAPTIONING

RNN



CNN



test image

before:

h = tanh(Wxh * x + Whh * h)

now:

h = tanh(Wxh * x + Whh * h + **Wih** * v)

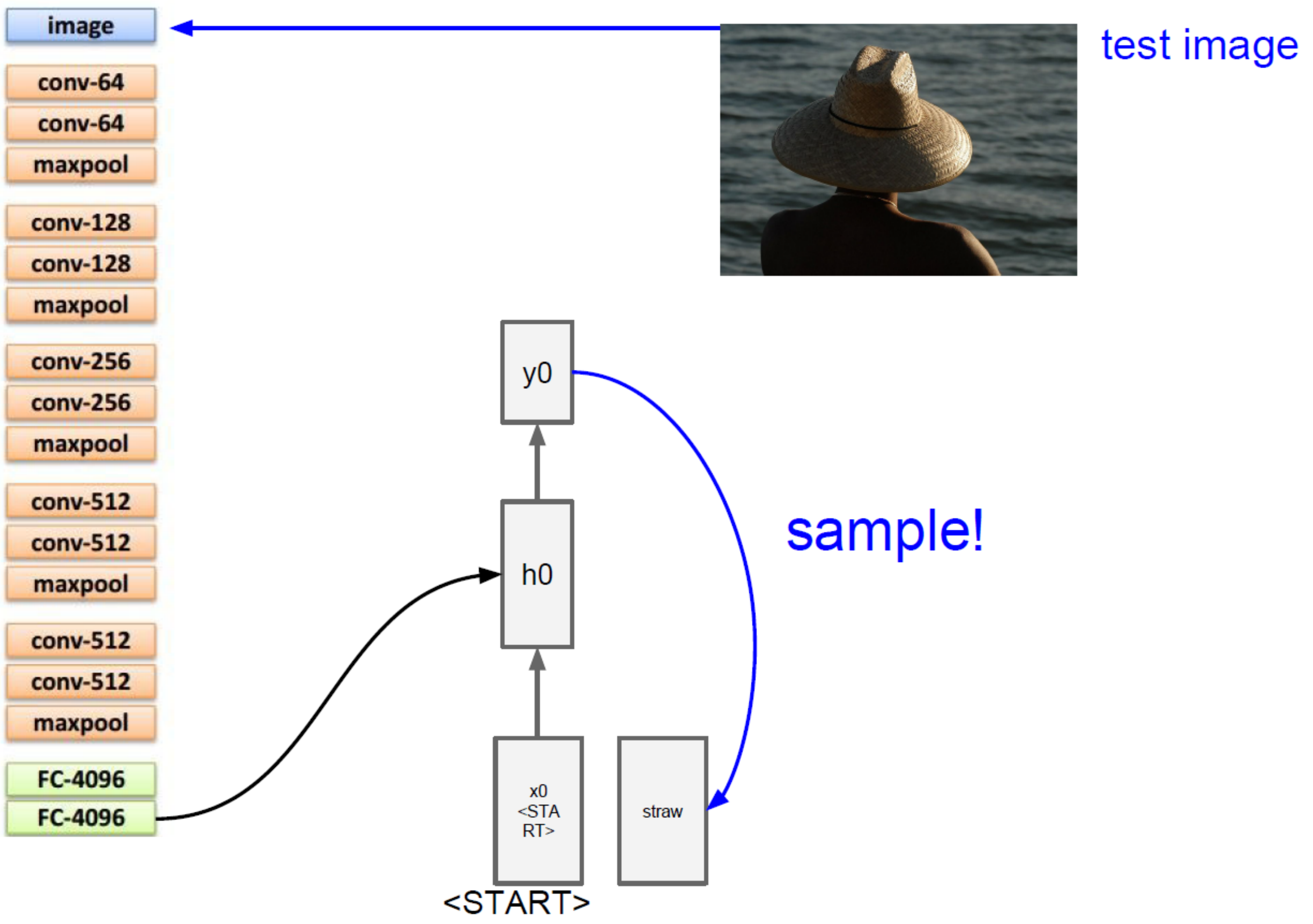


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
test image
y0
h0
sample!
x0 <STA RT>
straw
<START>

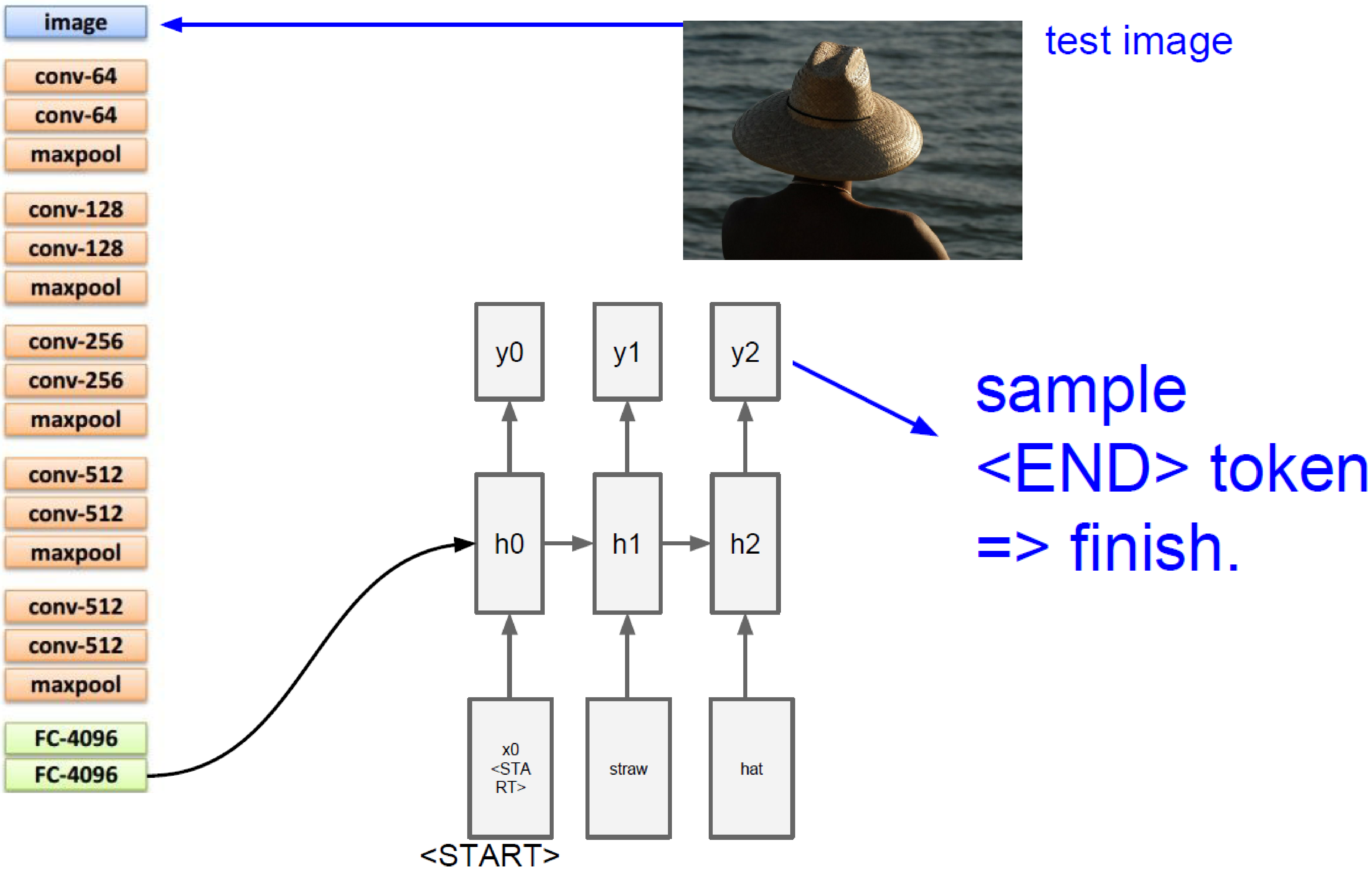


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
test image
y0
y1
y2
h0
h1
h2
x0 <START>
straw
hat
<START>
sample <END> token => finish.

IMAGE CAPTIONING

- Supervised learning.
- 데이터는 natural language cation이 있는 이미지(microsoft coco dataset)
- Train data 와 유사한 이미지일 수록 잘 작동.


A cat sitting on a suitcase on the floor


A cat is sitting on a tree branch


A dog is running in the grass with a frisbee


A white teddy bear sitting in the grass


Two people walking on the beach with surfboards


A tennis player in action on the court

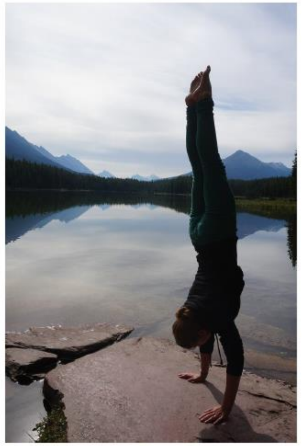

Two giraffes standing in a grassy field


A man riding a dirt bike on a dirt track

<success>


A woman is holding a cat in her hand


A person holding a computer mouse on a desk


A woman standing on a beach holding a surfboard

<failure>

IMAGE CAPTIONING WITH ATTENTION

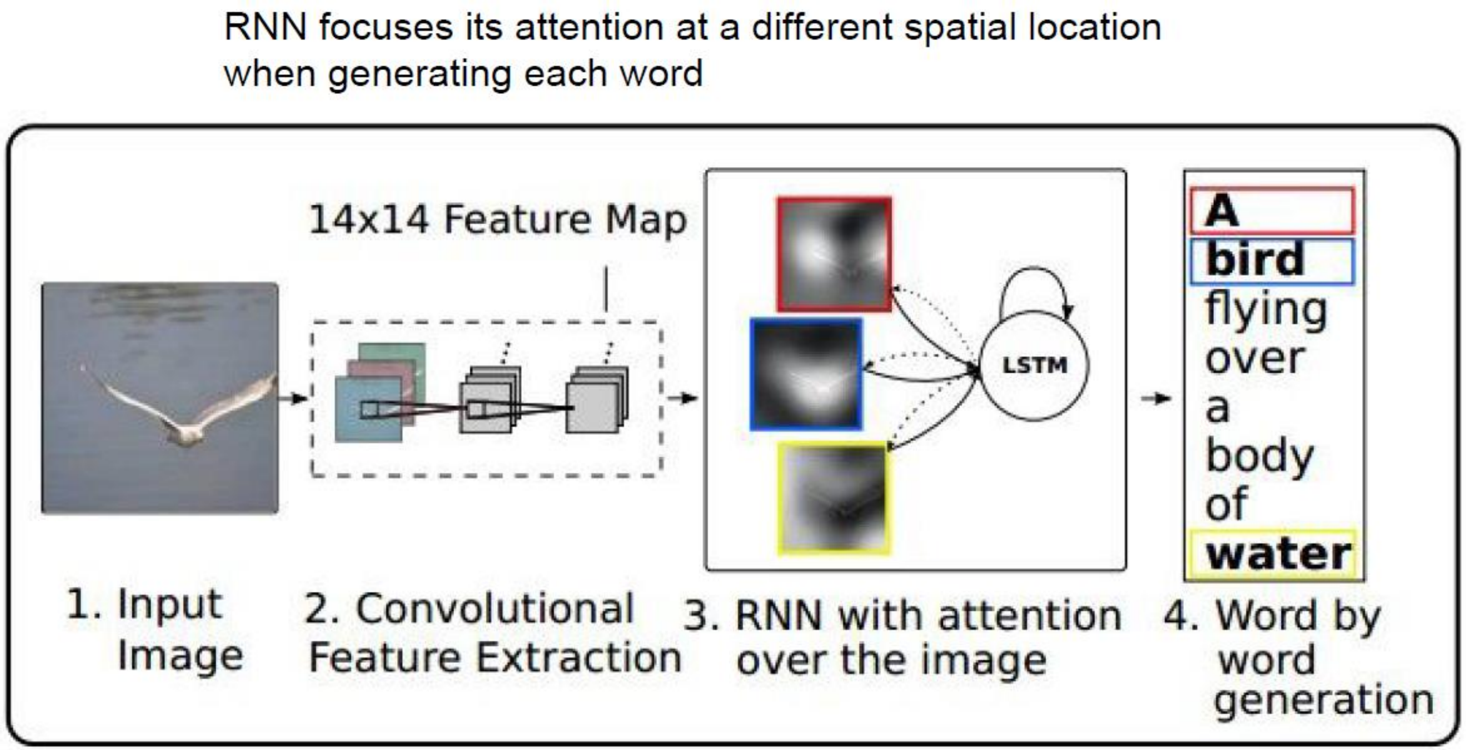
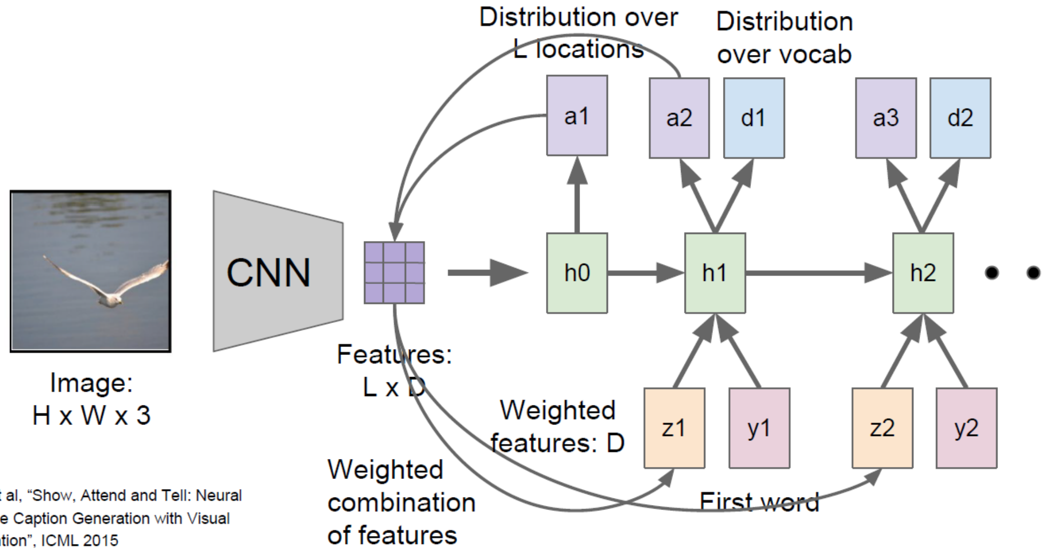
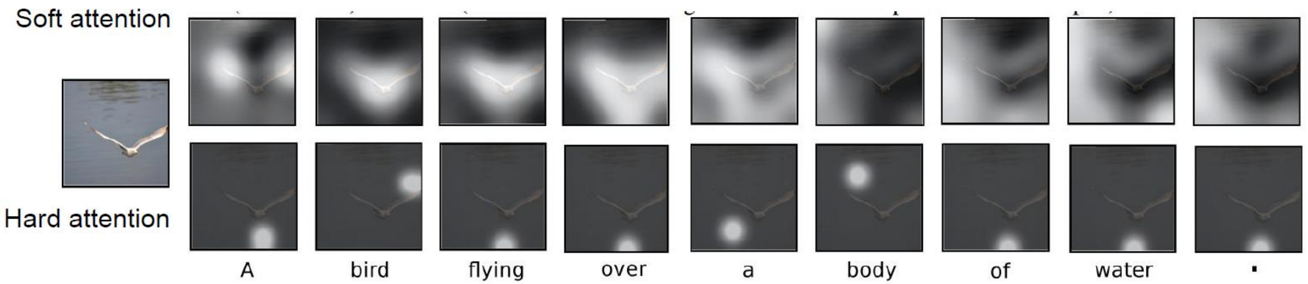


IMAGE CAPTIONING WITH ATTENTION

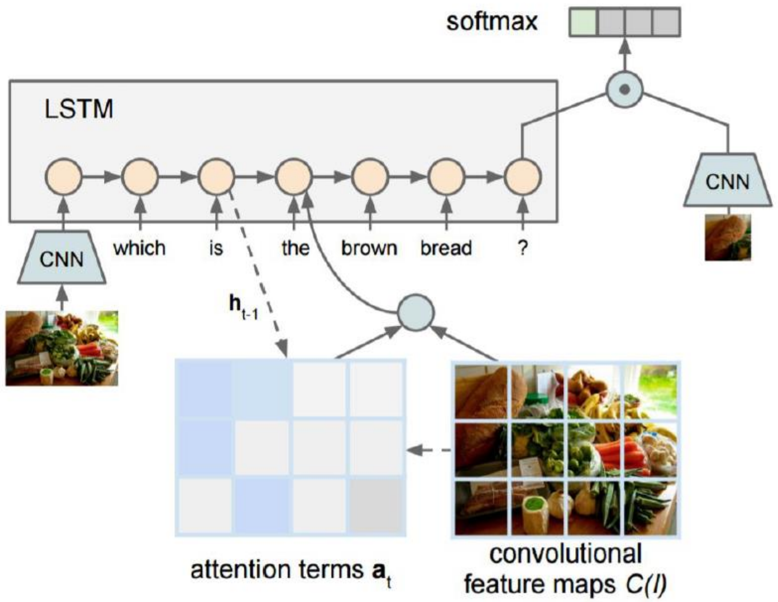


Xu et al, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention", ICML 2015



단어의 기반하여 특정 부분을 강조.(Attention)

VISUAL QUESTION ANSWERING



al, "Visual 7W: Grounded Question Answering in Images", CVPR 2016

MULTILAYER RNNS

Multilayer RNNs

$$\boxed{h_t^l = \tanh W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}}$$

$h \in \mathbb{R}^n$ $\quad W^l \ [n \times 2n]$

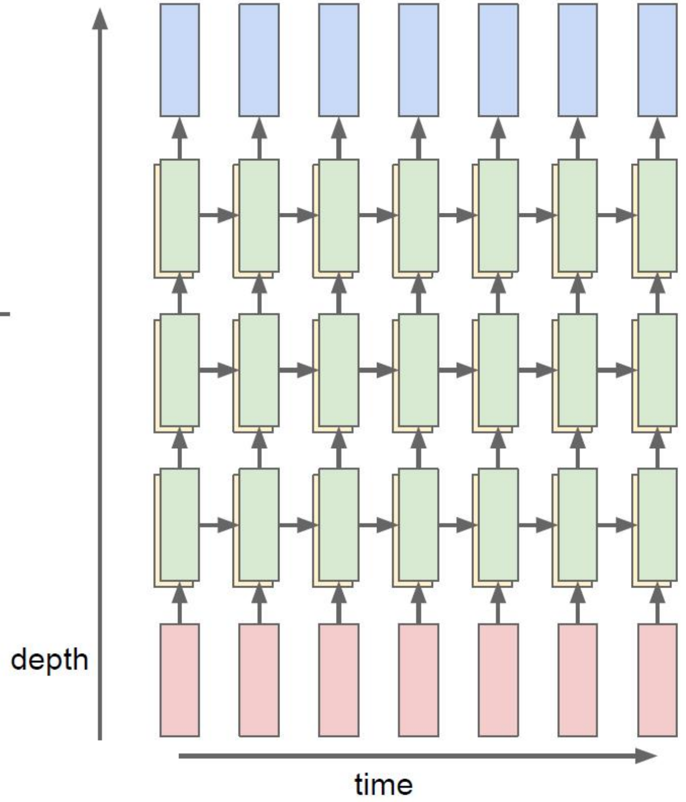
LSTM:

$W^l \ [4n \times 2n]$

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$

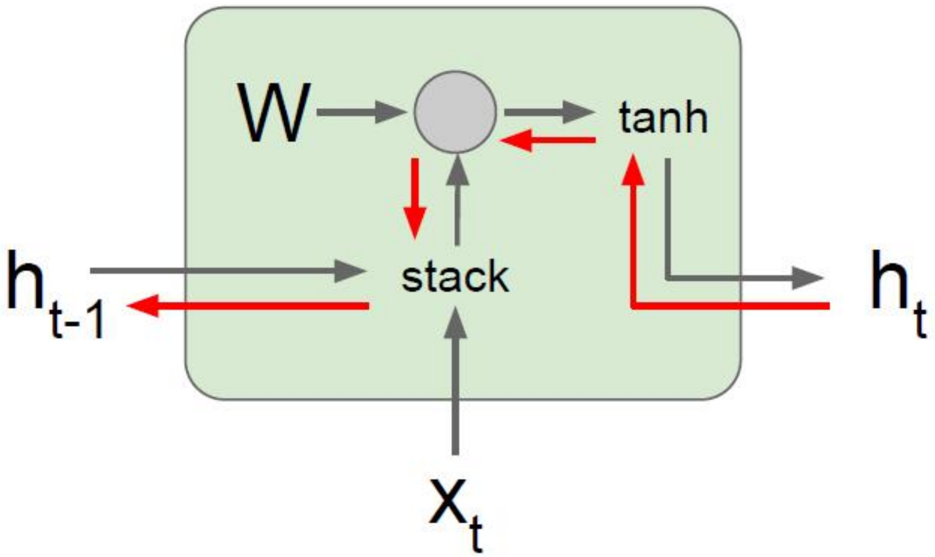
$$c_t^l = f \odot c_{t-1}^l + i \odot g$$

$$h_t^l = o \odot \tanh(c_t^l)$$



- CNN처럼 layer를 쌓을 수 있고 쌓을 수록 더 좋은 성능을 낸다
- 2~4 layer가 적당하다.

VANILLA RNN GRADIENT FLOW



$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$
$$= \tanh\left(\begin{pmatrix} W_{hh} & W_{hx} \end{pmatrix} \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right)$$
$$= \tanh\left(W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right)$$

```
grad_norm = np.sum(grad * grad)
if grad_norm > threshold:
  grad *= (threshold / grad_norm)
```

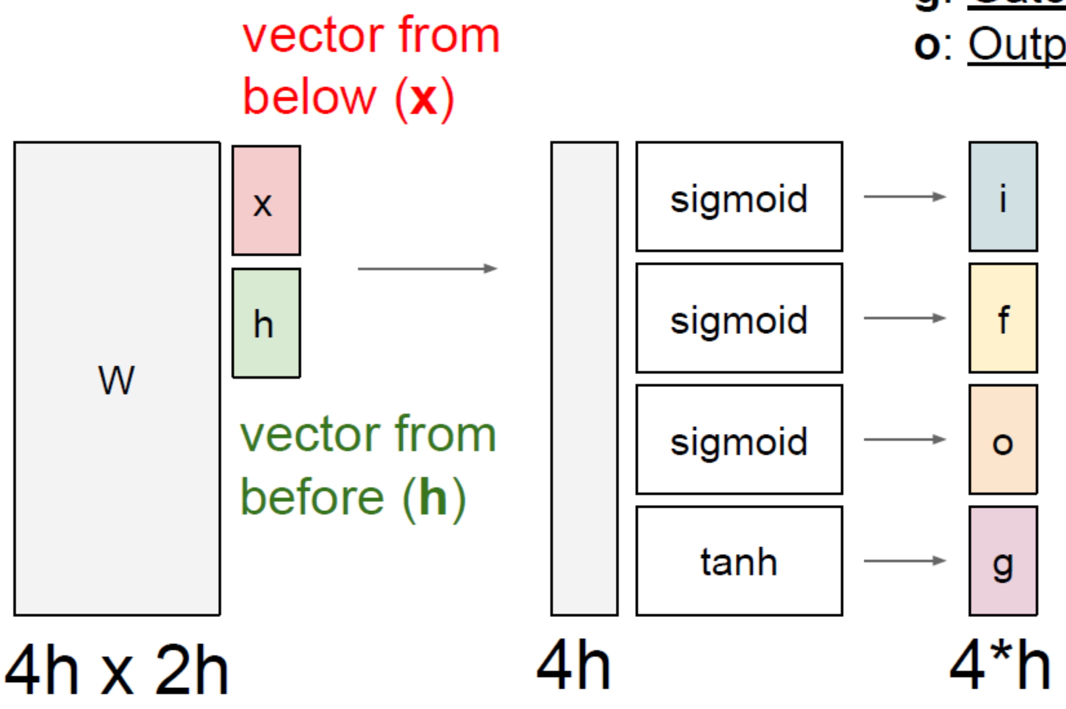
L2 임계값보다 크면 나눠줌

W_{hh}^T 가 계속 곱해지게 됨
→ Factor 가 1보다 크면 발산 작으면 0으로 수렴해버림(커지는 것은 clipping 으로 scale 을 줄일수 있으나 작아지는건 도리가 없음)

Long Short Term Memory (LSTM)

[Hochreiter et al., 1997]

f: <u>Forget gate</u>, Whether to erase cell
i: <u>Input gate</u>, whether to write to cell
g: <u>Gate gate</u> (?), How much to write to cell
o: <u>Output gate</u>, How much to reveal cell

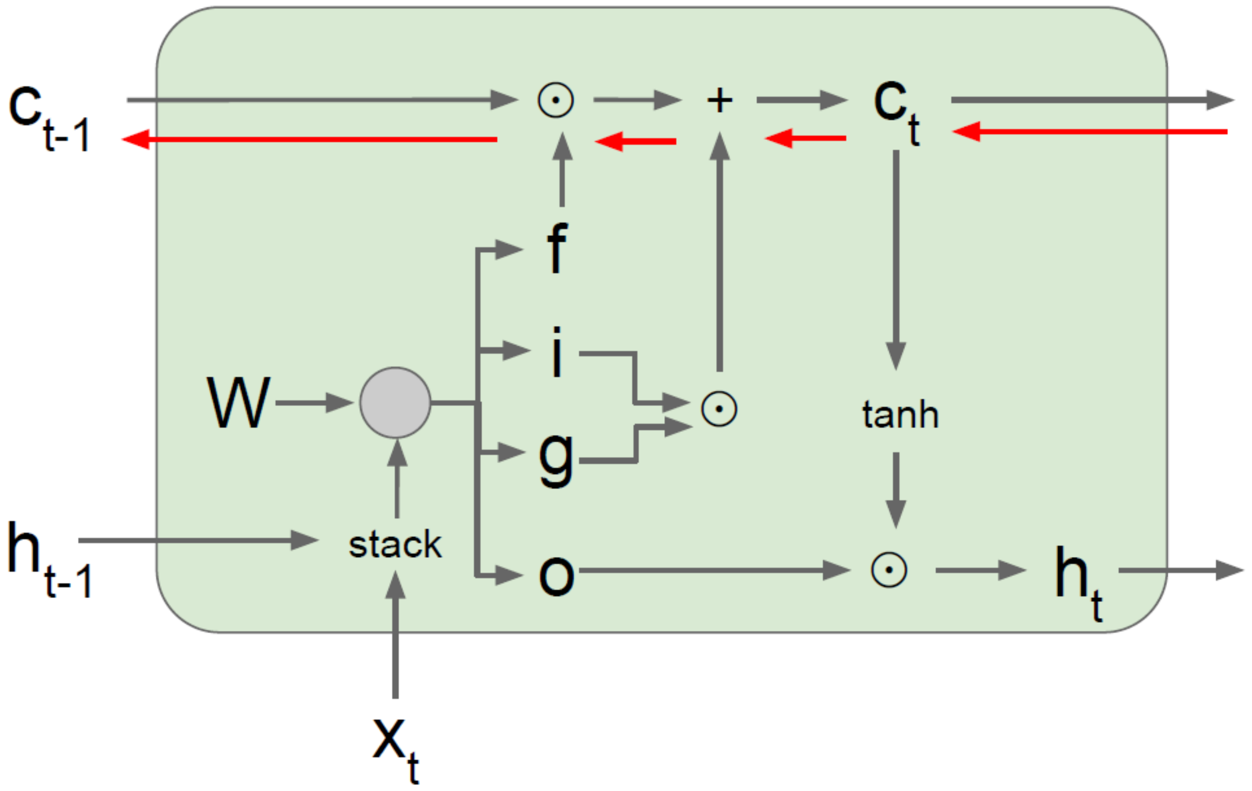


$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

$$c_t = f \odot c_{t-1} + i \odot g$$

$$h_t = o \odot \tanh(c_t)$$

LSTM



Backpropagation from c_t to c_{t-1} only elementwise multiplication by f, no matrix multiply by W

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

$$c_t = f \odot c_{t-1} + i \odot g$$

$$h_t = o \odot \tanh(c_t)$$

OTHER RNN

Other RNN Variants

GRU [*Learning phrase representations using rnn encoder-decoder for statistical machine translation*, Cho et al. 2014]

$$r_t = \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r)$$

$$z_t = \sigma(W_{xz}x_t + W_{hz}h_{t-1} + b_z)$$

$$\tilde{h}_t = \tanh(W_{xh}x_t + W_{hh}(r_t \odot h_{t-1}) + b_h)$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t$$

[*LSTM: A Search Space Odyssey*, Greff et al., 2015]

[*An Empirical Exploration of Recurrent Network Architectures*, Jozefowicz et al., 2015]

MUT1:

$$\begin{aligned} z &= \mathrm{sigm}(W_{\mathrm{xz}}x_t + b_{\mathrm{z}}) \\ r &= \mathrm{sigm}(W_{\mathrm{xr}}x_t + W_{\mathrm{hr}}h_t + b_{\mathrm{r}}) \\ h_{t+1} &= \tanh(W_{\mathrm{hh}}(r \odot h_t) + \tanh(x_t) + b_{\mathrm{h}}) \odot z \\ &+ h_t \odot (1 - z) \end{aligned}$$

MUT2:

$$\begin{aligned} z &= \mathrm{sigm}(W_{\mathrm{xz}}x_t + W_{\mathrm{hz}}h_t + b_{\mathrm{z}}) \\ r &= \mathrm{sigm}(x_t + W_{\mathrm{hr}}h_t + b_{\mathrm{r}}) \\ h_{t+1} &= \tanh(W_{\mathrm{hh}}(r \odot h_t) + W_{xh}x_t + b_{\mathrm{h}}) \odot z \\ &+ h_t \odot (1 - z) \end{aligned}$$

MUT3:

$$\begin{aligned} z &= \mathrm{sigm}(W_{\mathrm{xz}}x_t + W_{\mathrm{hz}}\tanh(h_t) + b_{\mathrm{z}}) \\ r &= \mathrm{sigm}(W_{\mathrm{xr}}x_t + W_{\mathrm{hr}}h_t + b_{\mathrm{r}}) \\ h_{t+1} &= \tanh(W_{\mathrm{hh}}(r \odot h_t) + W_{xh}x_t + b_{\mathrm{h}}) \odot z \\ &+ h_t \odot (1 - z) \end{aligned}$$

LSTM 이 낫다.

THANKS