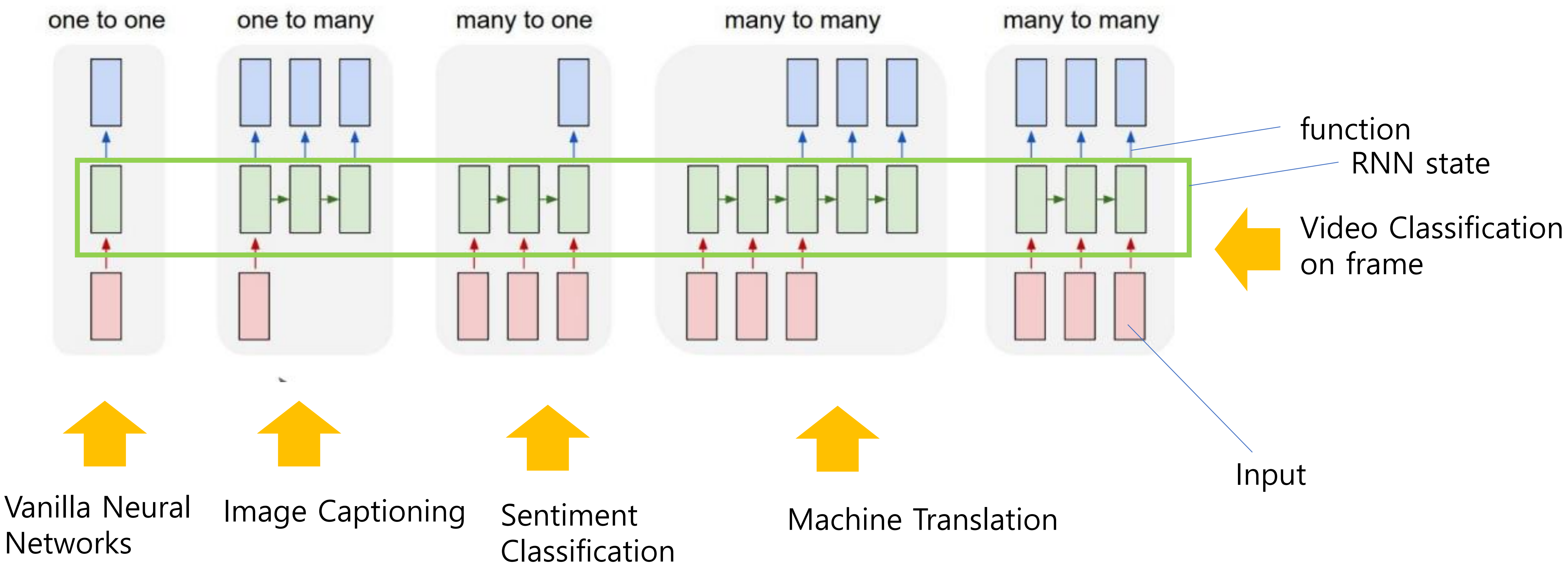
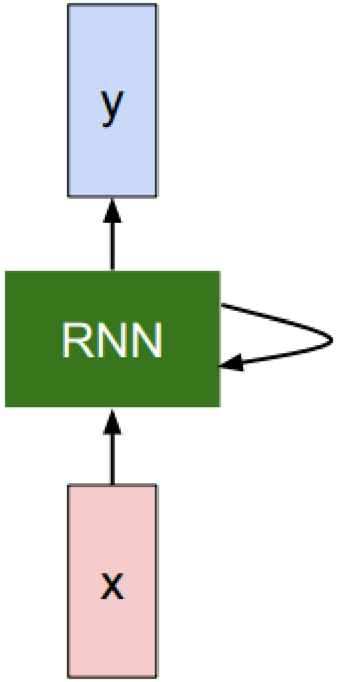
Cs231n 10주차
Recurrent Neural Networks
(RNN)

Recurrent Neural Networks: Process Sequences



Recursive Neural Networks (RNN)



- Recurrent Core cell이 있음
- input과 output을 RNN이 결합하여 새로운 state 만들어냄
- Hidden state에서 새로운 입력 받아들여 update
- 출력 값 출력

Sequential processing
of non-sequence data

We can process a sequence of vectors **x** by applying a **recurrence formula** at every time step:

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

new state
old state
input vector at some time step
some function with parameters W

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

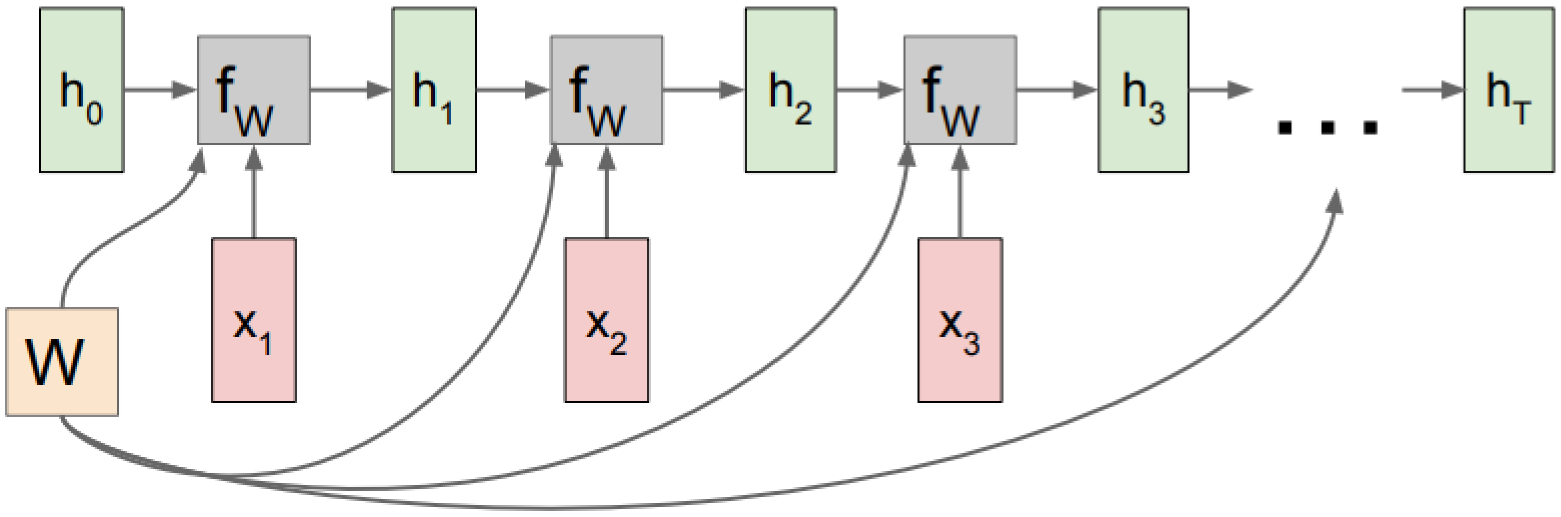
Vanila RNN 일경우 ->

$$y_t = W_{hy}h_t$$

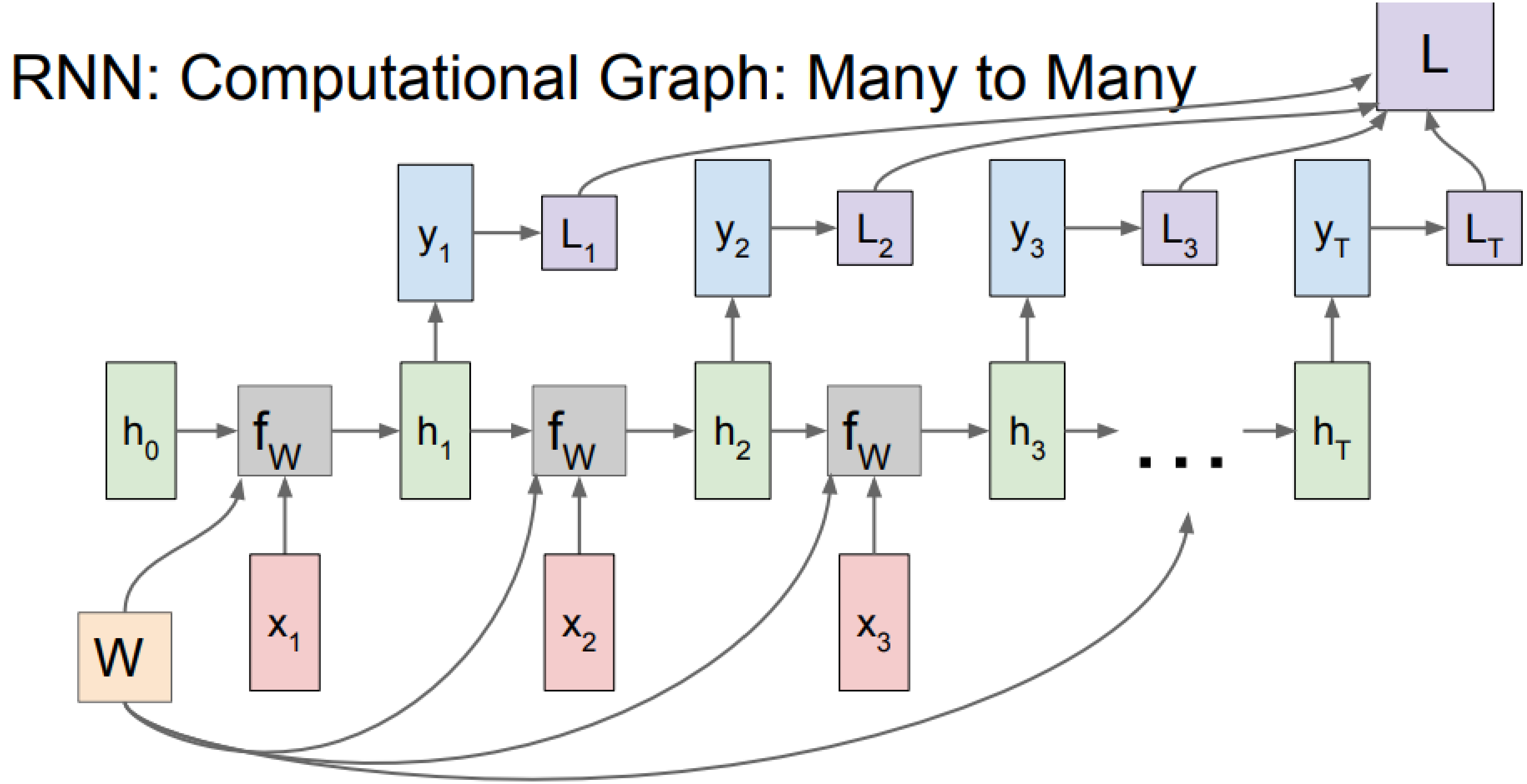
RNN 작동방식: Computational Graph

- *fw*(*h*0,*x*1)+W gradient->*fw*(*h*01,*x2*) + W gradient…

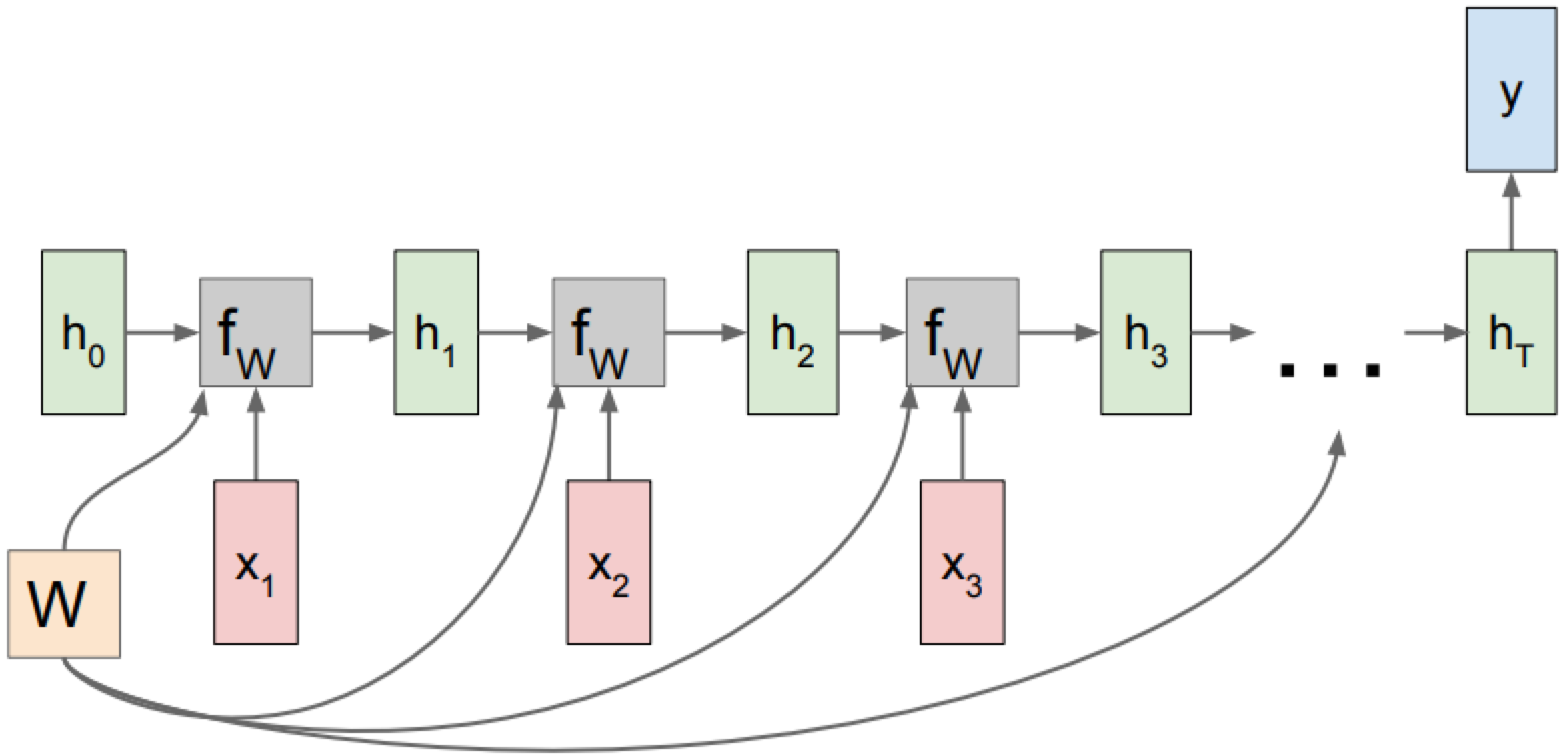
Re-use the same weight matrix at every time-step



- *h*가 y의 입력으로 들어감

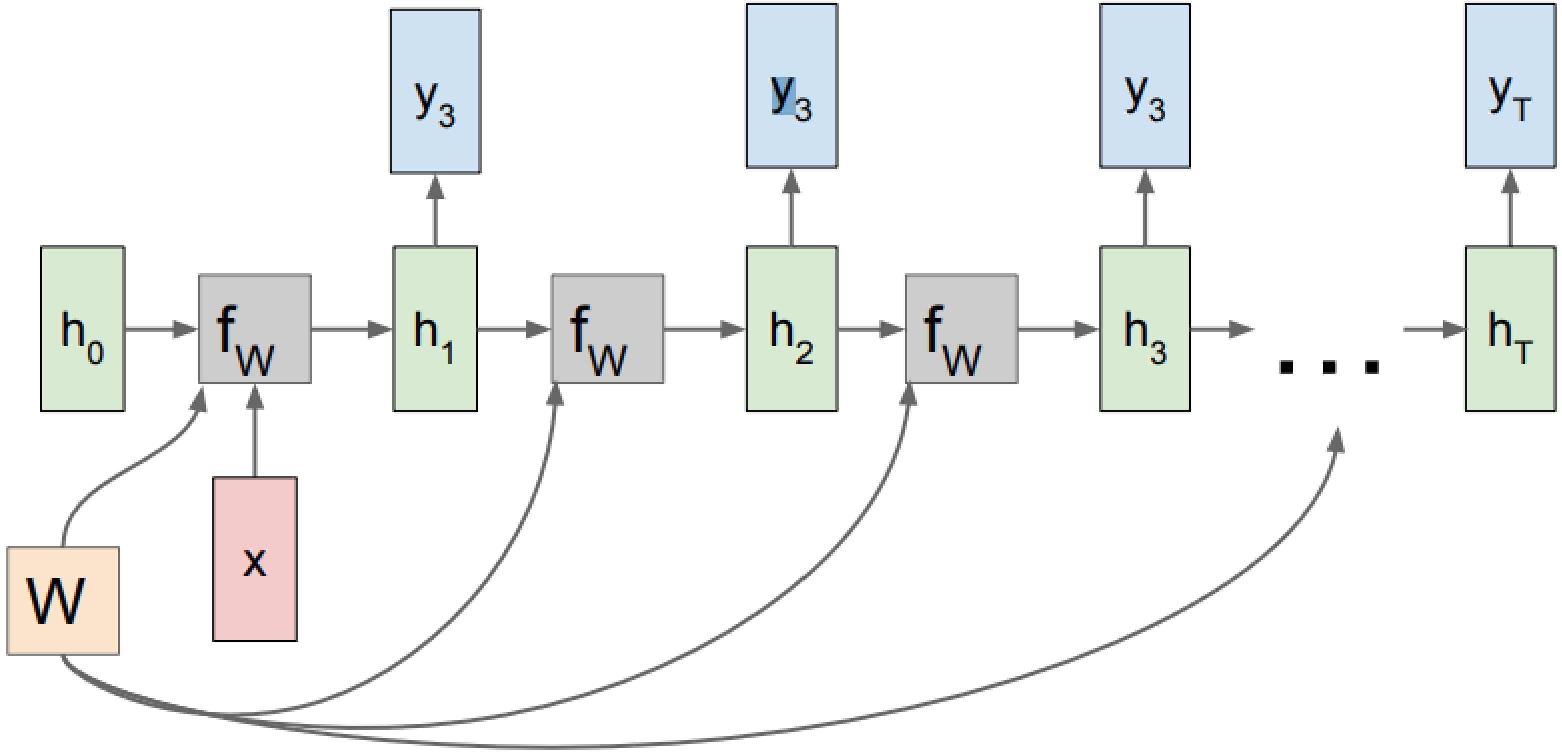


RNN: Computational Graph: Many to One



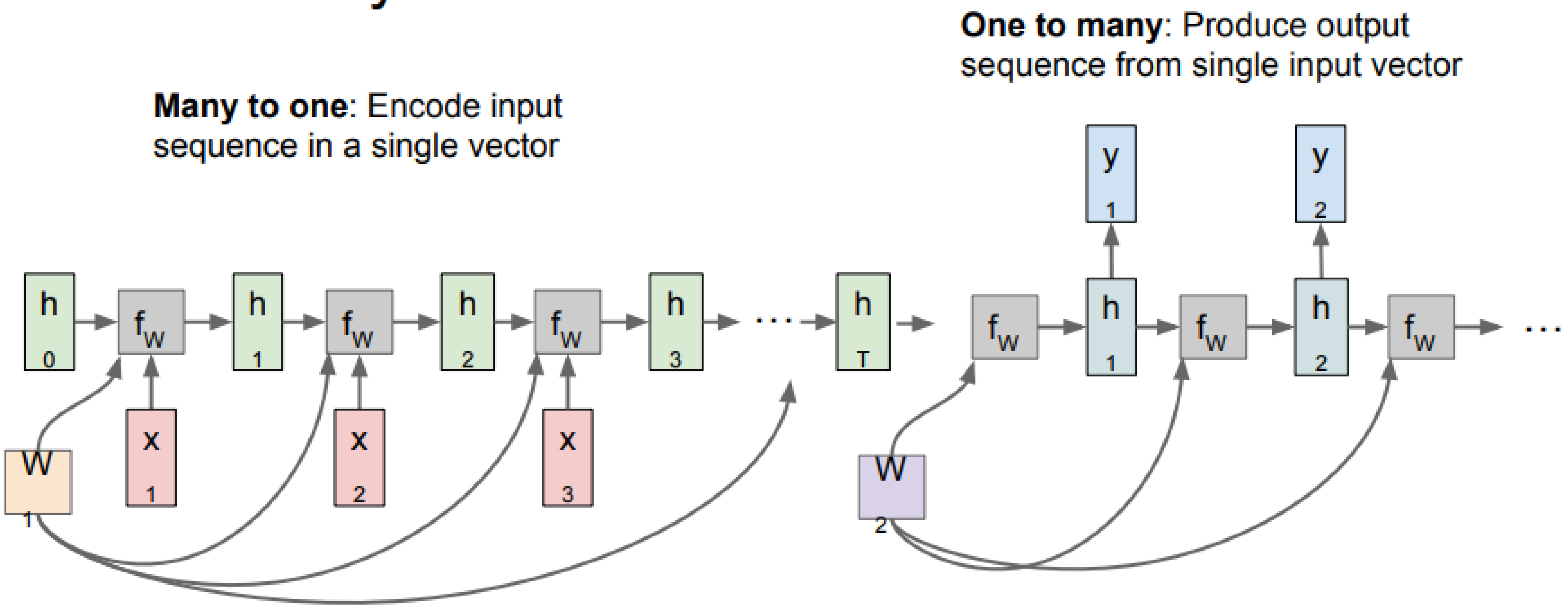
- 감정 분석(sentiment analysis에 쓰임
- 최종 hidden state가 전체 sequence에 대한 요약

RNN: Computational Graph: One to Many



- 고정 input & 가변 output

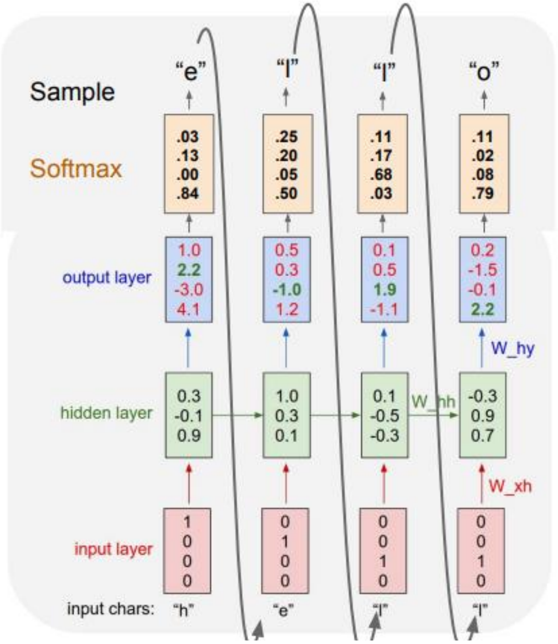
Sequence to Sequence: Many-to-one + one-to-many

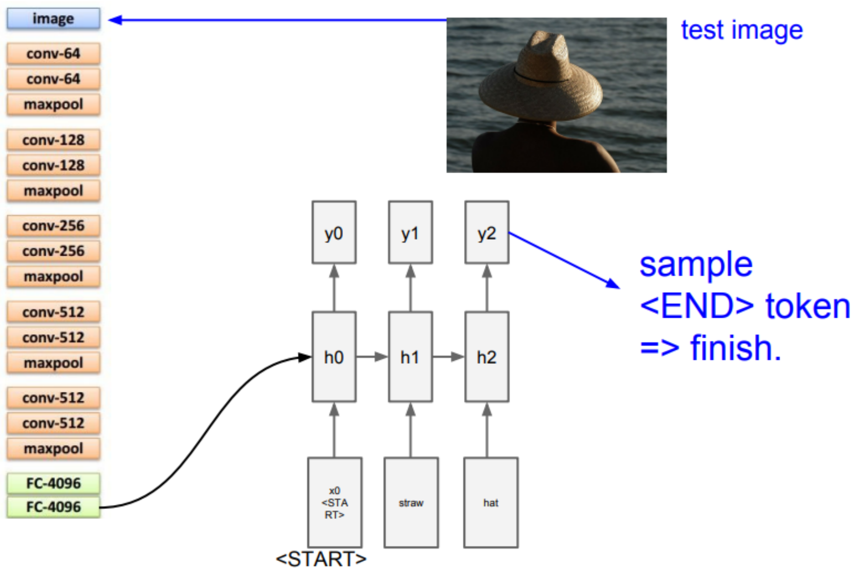


- Machine translation에 쓰임
- encoder+decoder

RNN: Natural Language Model

- Character-level Language Model
- Latent structure (Searching for interpretable cells
- Image captioning





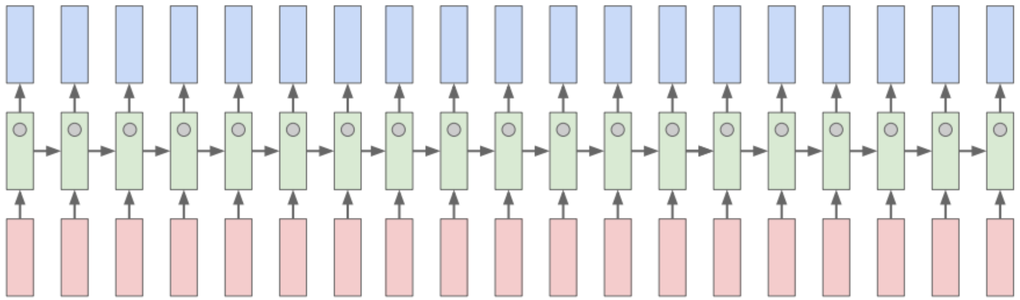
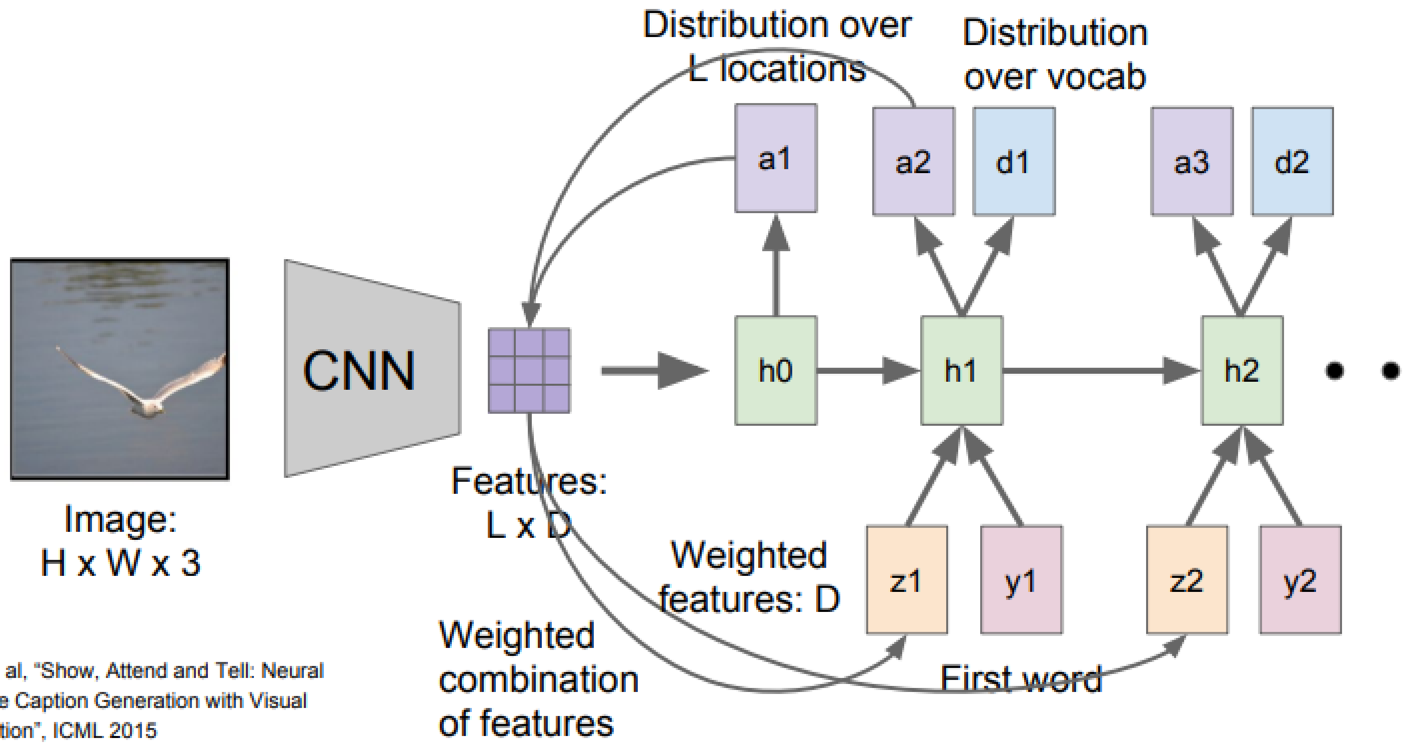


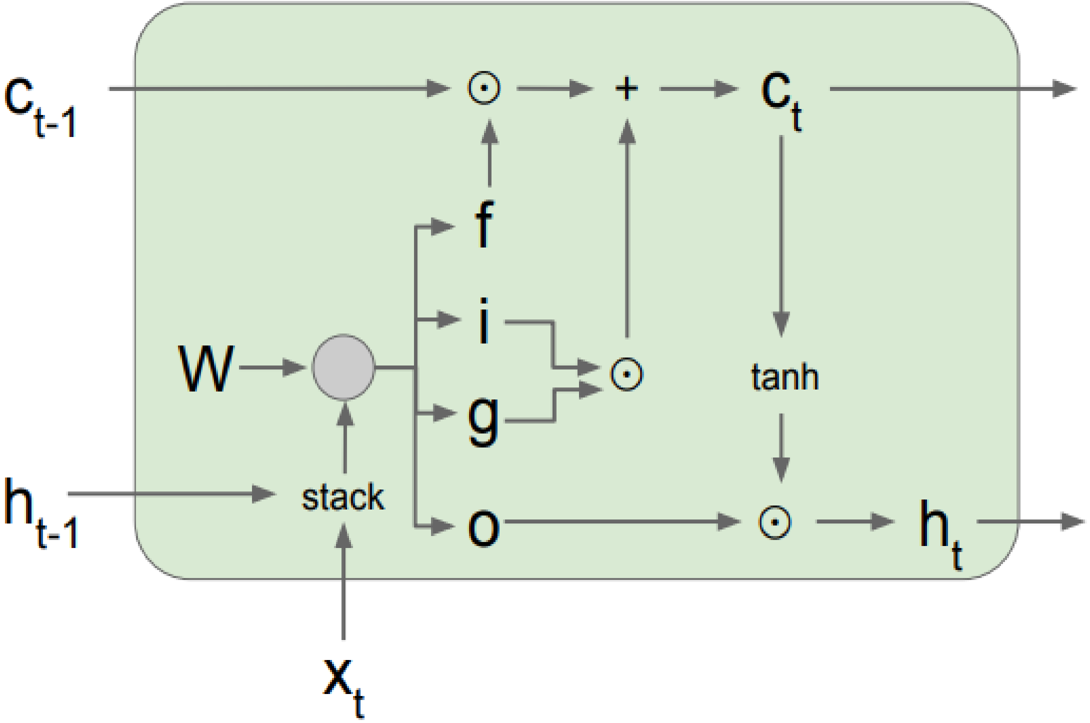
Image Captioning with Attention



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

- Forward pass시 매 step 모델이 이미지에서 보고싶은 image에 대한 분포 생성
- 이미지의 각 위치에 대한 분포= Train time에 모델이 어느 위치에 봐야하는 Attention

Long short Term Memory(LSTM)



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

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

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

- Input gate
- Forget gate
- Output gate
- Gate gate

RNN과 비교하였을때 장점
1.F와 곱해지는 연산이 matrix multiplication이 아닌 element-wise임
full matrix multiplication 보다 element wise multiplication이 더 나음
2.Element wise multiplication을 통해 매 스텝 다른 값의 f와 곱해질 수 있음
3.Vanishig gradient 덜 민감 (+GRU)