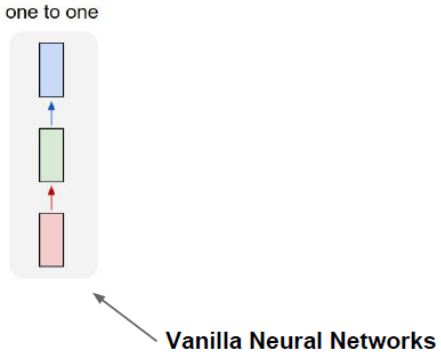
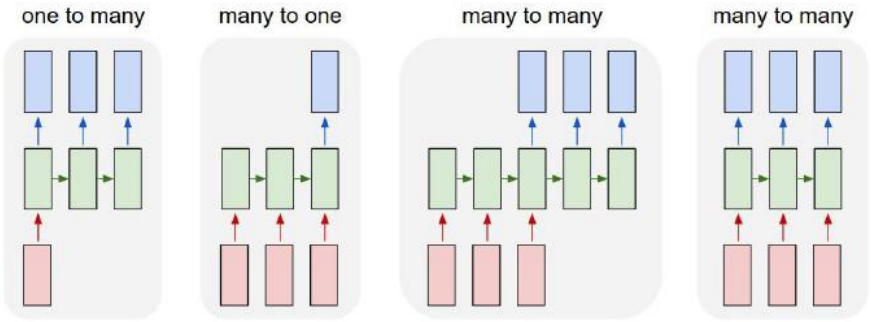
CS231n 10강 요약 CV팀 2071042 임연우

1. RNN



여태 배운 구조를 Vanilla Neural Network라 하자. 입력이 하나 들어가서 hidden layer를 거쳐서 출력이 하나인 구조이다.



RNN은 Recurrent Neural Network의 줄임말.

다양한 입출력을 다룰 수 있도록 만든 모델.

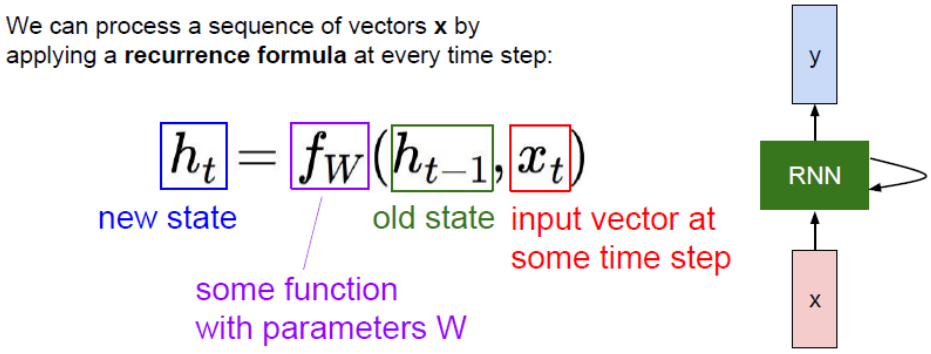
One to many : Image Captioning

Manty to one : sentiment classification

Many to many : machine translation, video classification on frame level

***** RNN의 흐름**

1) 입력을 받는다.
2) 모든 step마다 hidden state를 update한다.
3) 출력값을 내보낸다.



매 step마다 x_t가 들어오고 기존에 있던 h_t-1 과 x_t를 이용해 h_t를 update.

h_0은 보통 0으로 초기화.

"같은 set의 파라미터들을 가지고 매 step마다 update를 진행한다"는 점이 포인트.

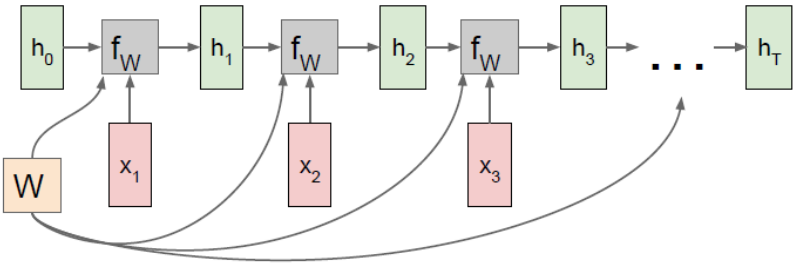
$$h_t = f_W(h_{t-1}, x_t)$$

$$\downarrow$$

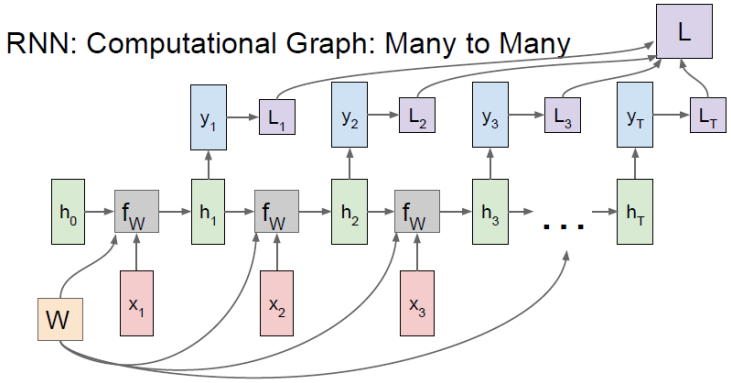
$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

$$y_t = W_{hy}h_t$$

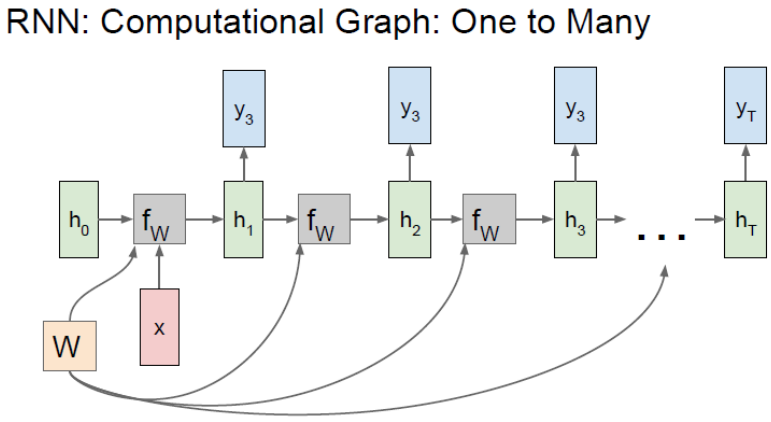
Re-use the same weight matrix at every time-step



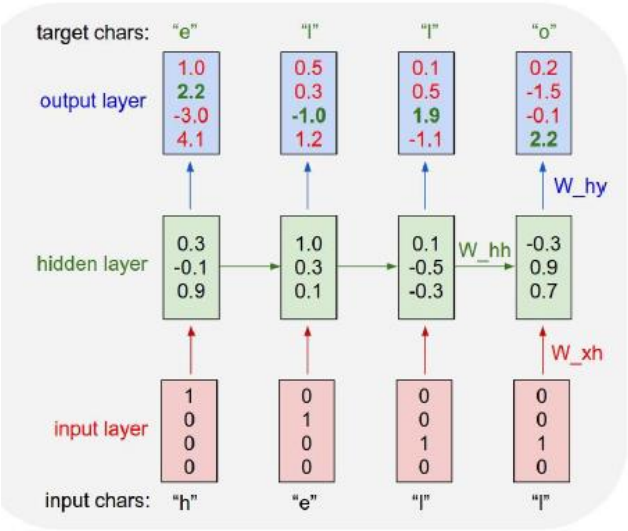
출력이 다수면 loss func도 여러 개.

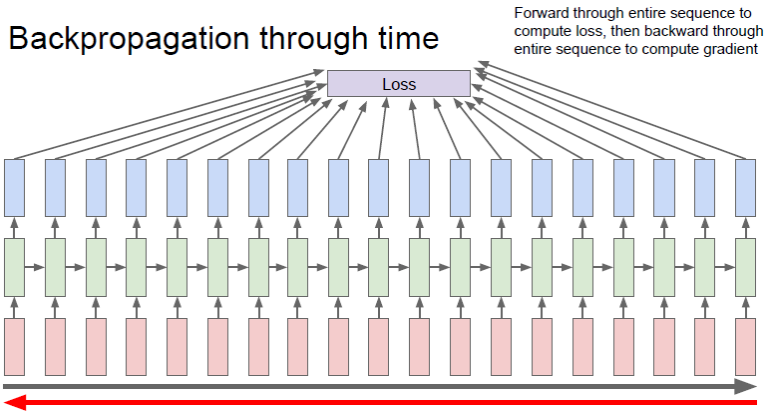


One to many면 computational graph는 다음과 같음.

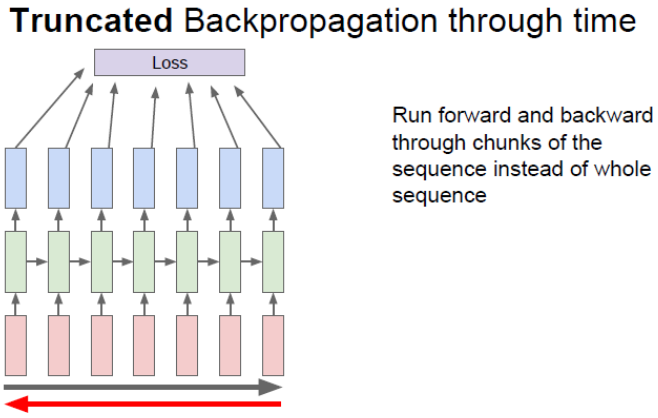


***** Character-level language model**



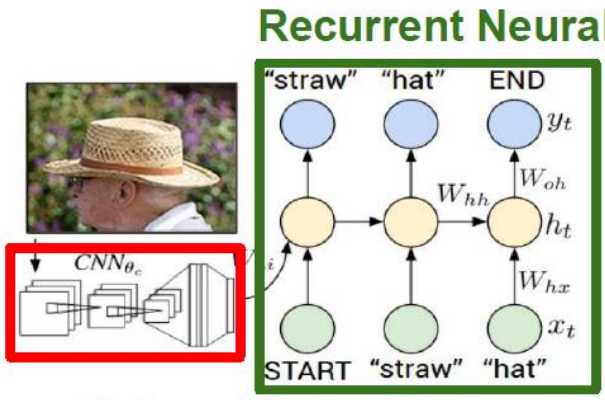


하나의 출력값을 알기위해 처음부터 모든 loss 값을 알아야하므로 위의 사진처럼 계산량이 매우 많다. 따라서 아래의 그림처럼 train할 때 한 step을 일정단위로 나누고 서브시퀀스의 loss들만 계산함.



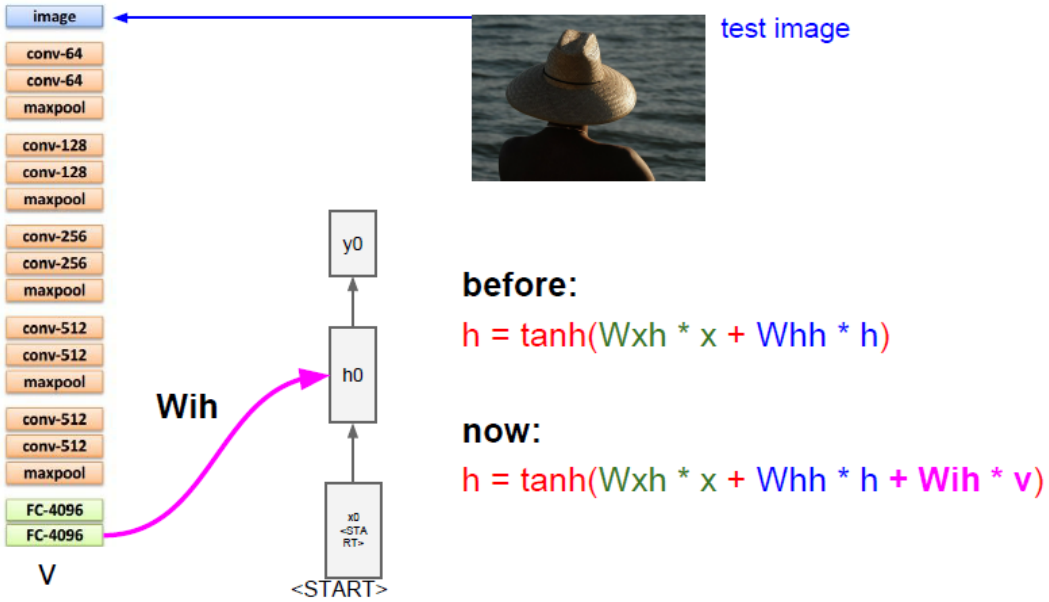
2. Image Captioning

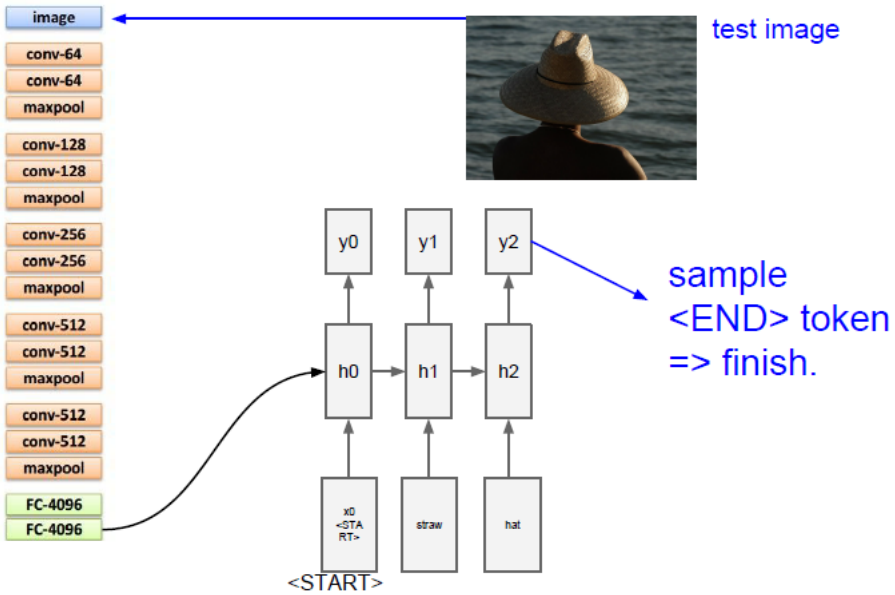
Image captioning = 하나의 사진을 보고 이 사진이 무엇인지 설명하는 문장을 만들어내는 딥러닝모델.



사진이 CNN구조를 통과하고, RNN구조를 통과한다.

test이미지가 CNN을 통과할 때 softmax를 통과하기 전에 FC layer를 이용한다.



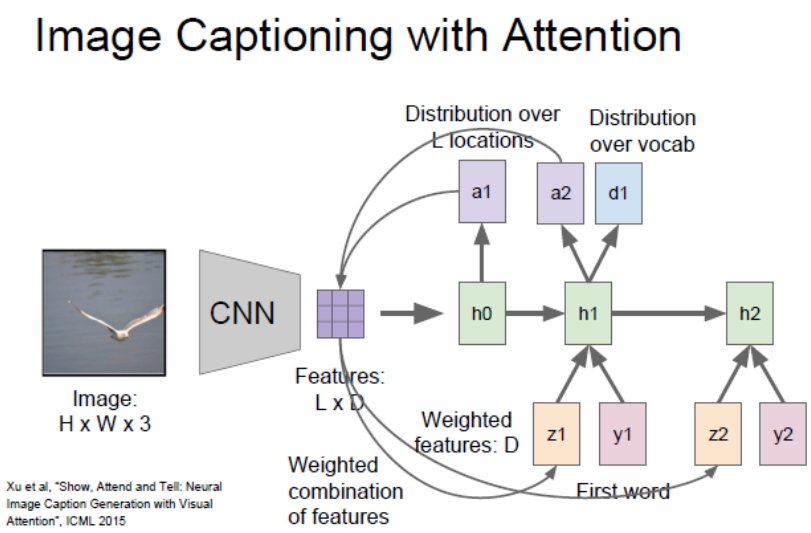


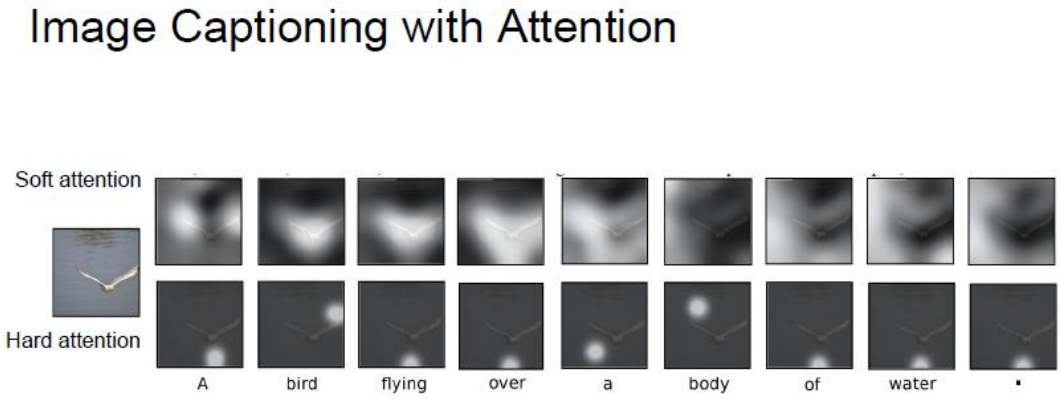
Supervised learning의 예시.

Supervised learning 보다 진보된 모델인 **image captioning with attention.**

Input image를 통해 공간정보를 담고 있는 grid of vector 값을 얻어낸다.

A1, a2 등이 이미지 분포, d1, d2 등이 단어 분포를 나타낸 출력값이다.





caption생성을 위해 이미지의 attention을 이동하는 모습. 의미가 있다고 생각하는 부분에 스스로 attention하는 모습.

Visual Question Answering (VQA)

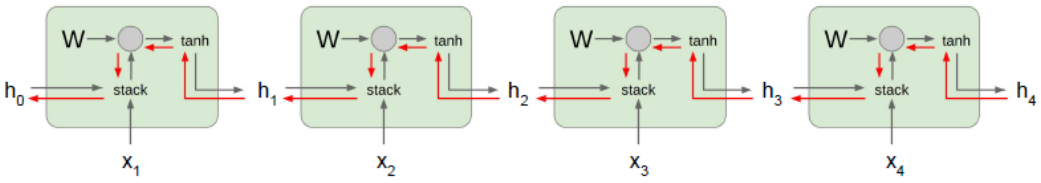
이미지와 질문을 동시에 던졌을 때 답을 맞추는 것.

이를 vanilla RNN gradient flow로 계산시,

RNN cell하나를 통과할때마다 가중치 행렬의 일부를 곱하게 됨.

따라서 많은 W 행렬들이 개입하게 되어 계산량이 늘어남. (단점1)

(단점2) weight 행렬에서 특이값의 최댓값에 따라 gradient값이 폭발적으로 증가하거나 너무 작게 감소할 수도 있음.



Computing gradient of h_0 involves many factors of W (and repeated tanh)

Largest singular value > 1: **Exploding gradients**

Largest singular value < 1: **Vanishing gradients**

3. LSTM

위의 문제를 해결하기위한 새로운 RNN 구조 = LSTM = Long Short Term Memory

Vanilla RNN

$$h_t = \tanh\left(W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right)$$

LSTM

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

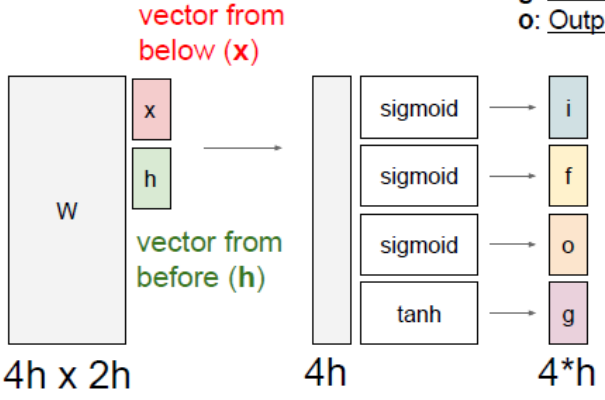
$$c_t = f \odot c_{t-1} + i \odot g$$

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

Long Short Term Memory (LSTM)

[Hochreiter et al., 1997]

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



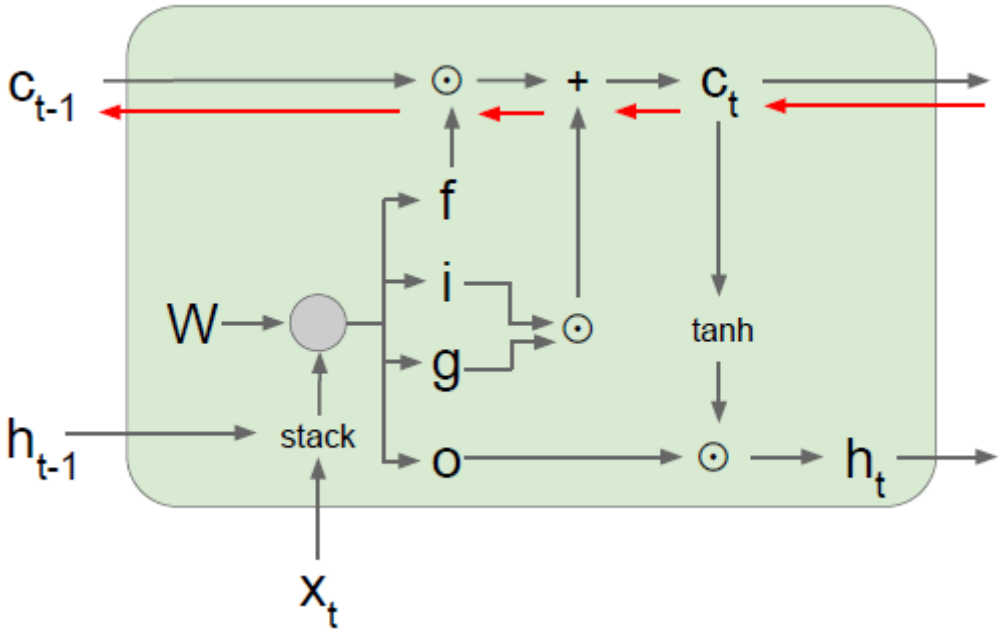
$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

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

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

4가지의 gate가 존재

- i - Input gate : cell 에 대한 입력 (X_t 에 대한 가중치)
- f - Forget gate : 이전 cell 에 대한 정보를 얼마나 지울 지
- o - Output gate : C_t 를 얼마나 밖으로 드러낼 것인지
- g - Gate gate(?) : Cell 을 얼마나 포함 시킬 것인지



하나의 cell에서 wieght행렬을 거치지않고도 cell단위에서 gradient계산이 가능 > 여러 cell을 거치더라도 계산량이 크게 증가하지 않음.

(forget gate 곱해지는 연산이 행렬곱아니고 원소곱(element-wise)라 계산량이 크게 증가하지 않음.