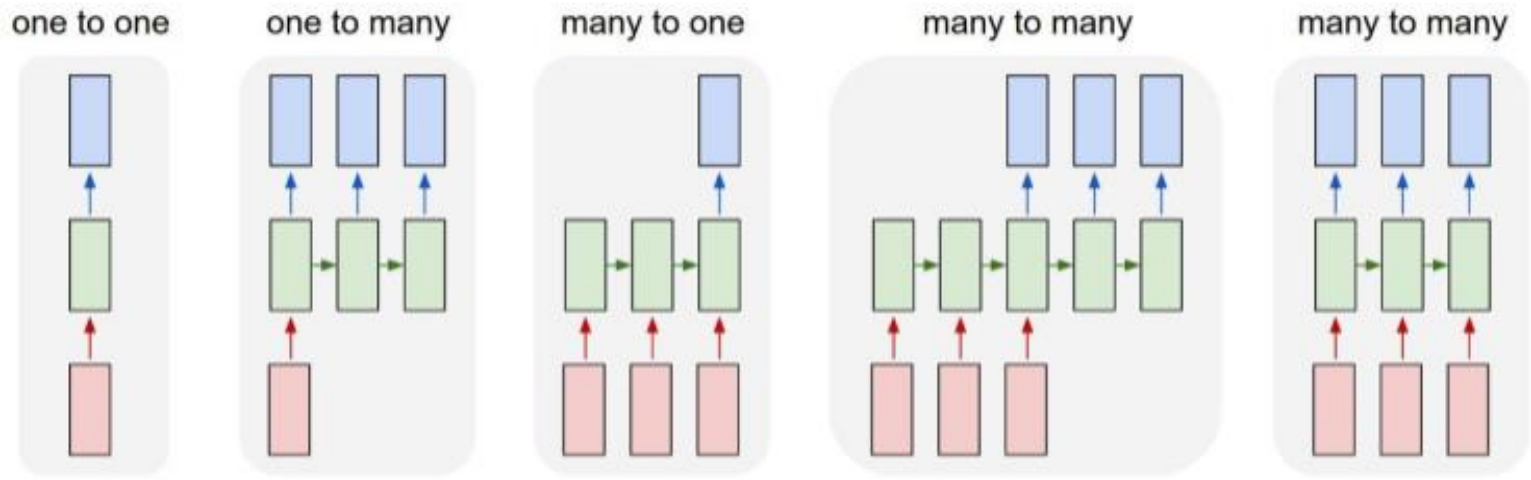
CS231n Lecture 10

- BOAZ 10기 박성현
- BOAZ 11기 김태희
- BOAZ 11기 홍지민
- BOAZ 10기 김용규

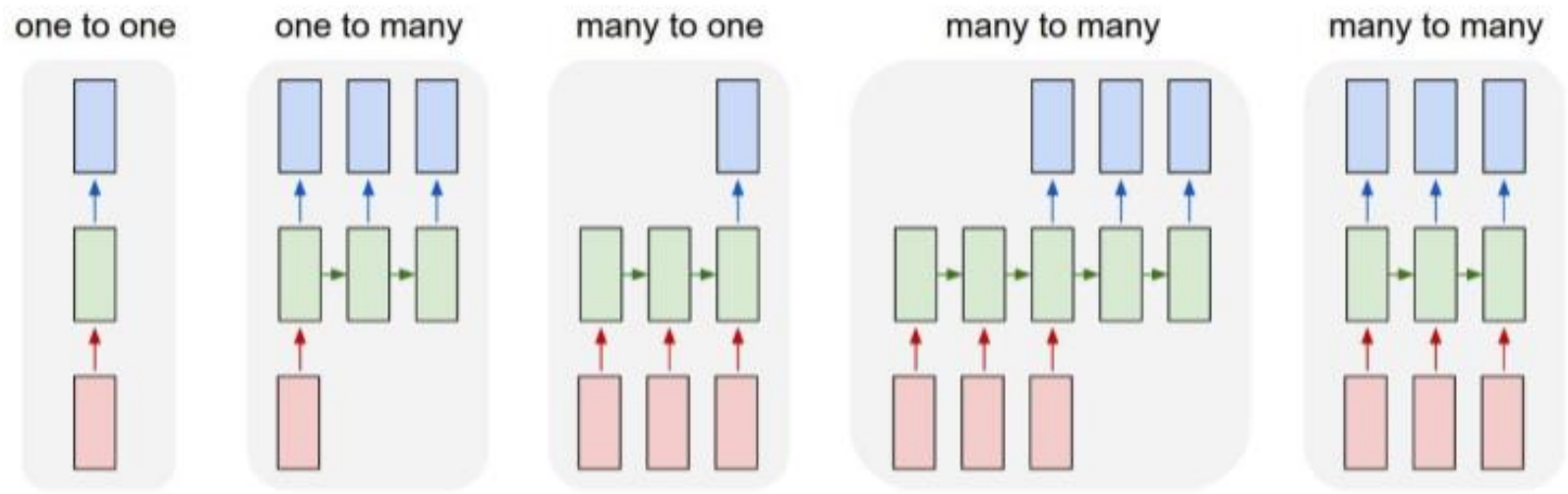
RNN

이때까지 배웠던 뉴럴넷과는 약간 다른 구조!

자기가 원하는 Task에 따라 다양한 길이의 output을 낼 수 있다. (flexible)



Recurrent Neural Networks



One to One – Fully connected Layer (이전의 인풋을 다음으로 한꺼번에 넘기는 구조)

One to Many - 하나의 인풋으로 여러개의 아웃풋을 생성 (image captioning)

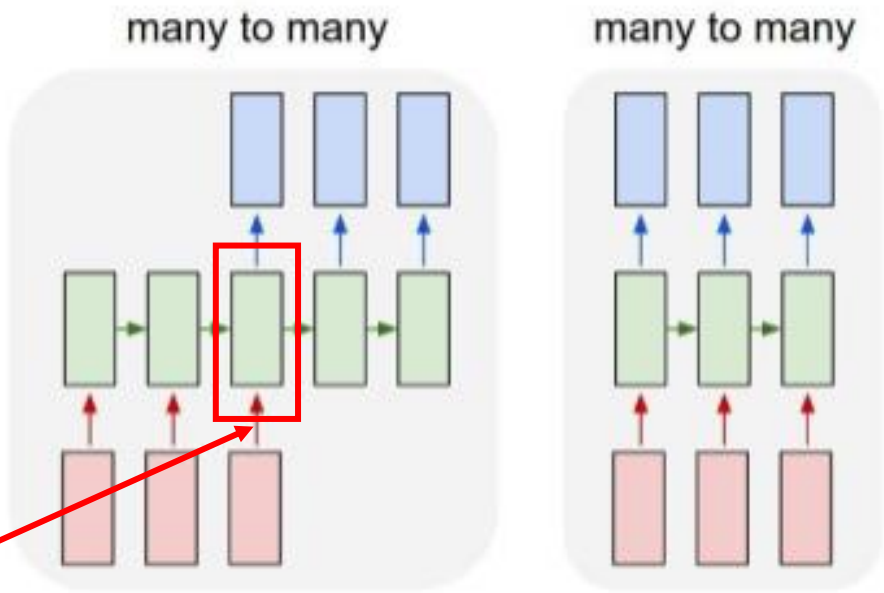
One to Many - 여러개의 인풋으로 하나의 아웃풋을 생성
(Sentiment Classification, 스팸 분류 등)

Many to Many - 여러개의 인풋으로 여러개의 아웃풋을 생성

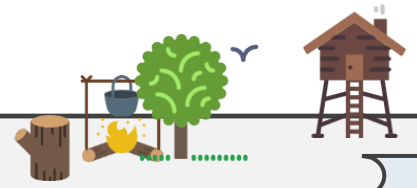
Recurrent Neural Networks

Many to many에서 중간의 벡터가 있는 case:
Seq2Seq 보통 NMT (기계번역) task에 사용

하나의 인풋이 들어올 때마다 바로 output이 나오는 case :
Language model 보통 Speech Recognition에서 사용됨.



이전 hidden state 정보들을 모두 반영했다고
해서 보통 문맥 벡터(Context vector)라 명명



Recurrent Neural Networks

RNN의 기본 구조

이전 hidden state와 현재 인풋의 조합으로 현재의 hidden state의 값이 나온다.

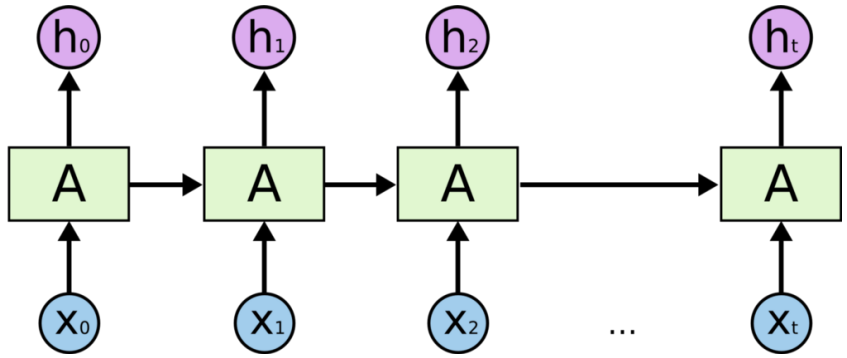
$$h_t = f_W(h_{t-1}, x_t)$$

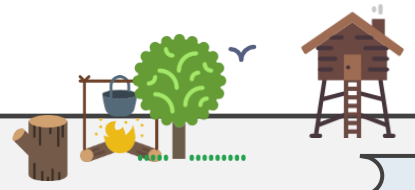
new state

old state

input vector at some time step

some function with parameters W





Recurrent Neural Networks

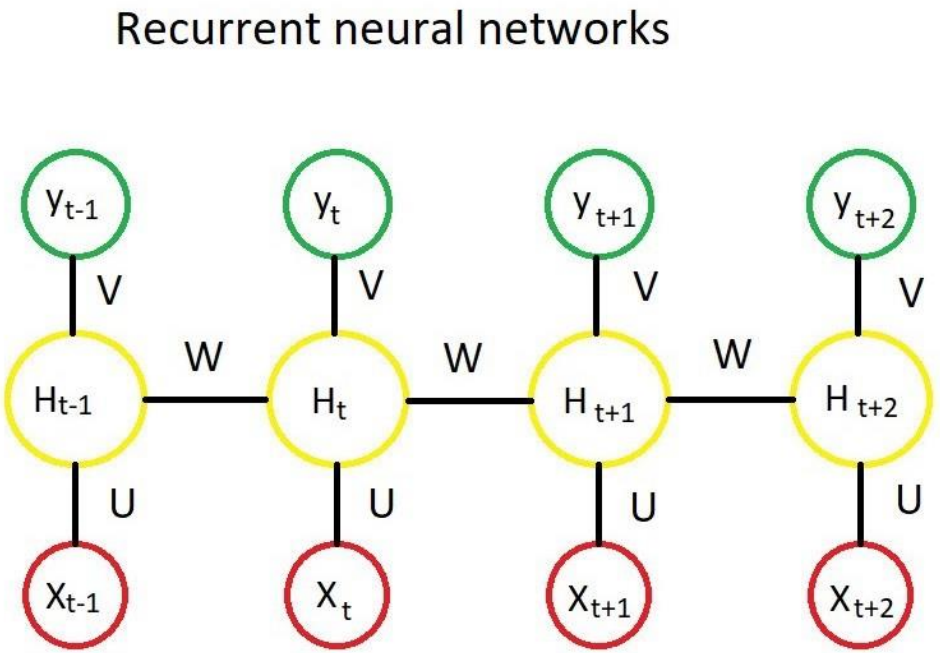
흥미로운 사실!
RNN은 모든 time step에서 weight matrix를 동일하게 적용!

왜 그럴까....?
들어가는 input의 자료형과 time step의 의미에 대해 생각해보기!

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

$$y_t = W_{hy}h_t$$

Weight Matrix는 단 세개!

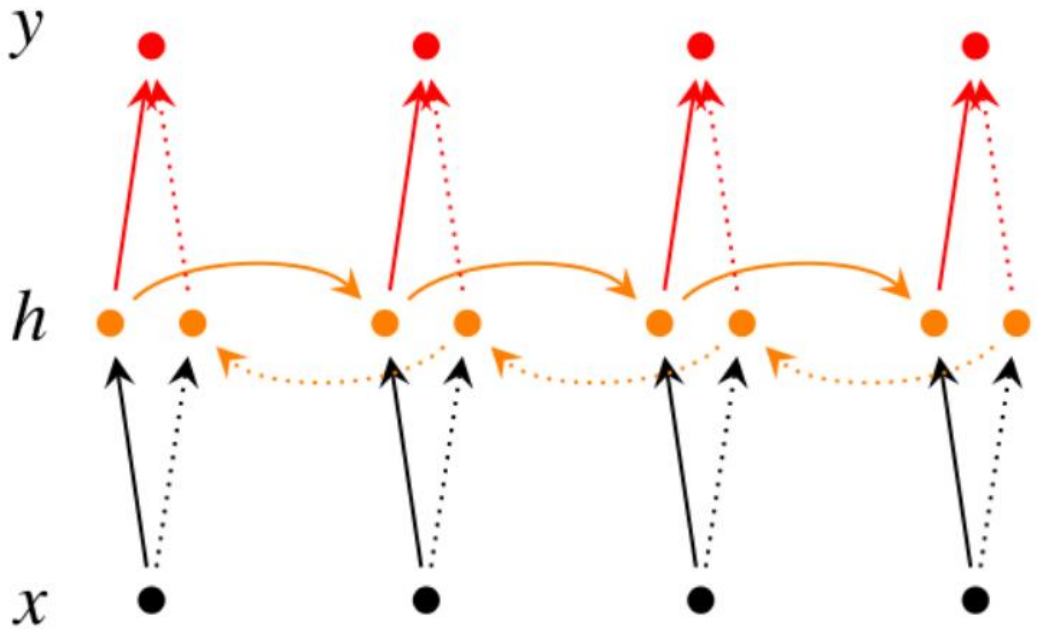


Bi – directional

이전 time step만 반영하는 것이 아닌 next step의 정보도 반영

Bi -directional을 쓸수 있는 구조
혹은 task면 반드시 써야 옳다!

쓸수 없는 Task는 뭐가 있을까..?

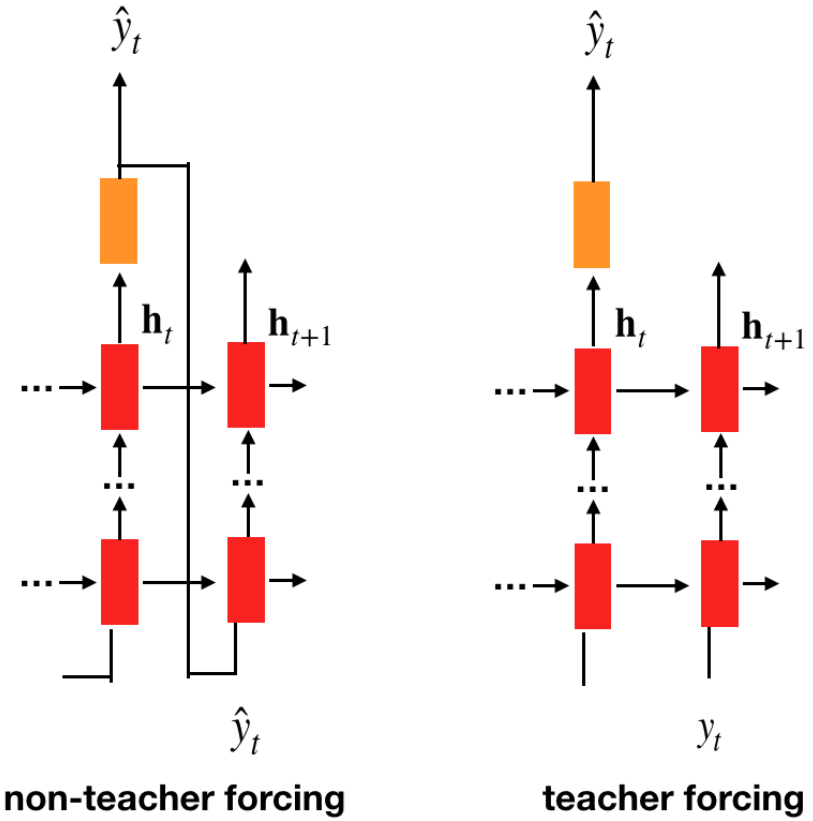


Teacher Forcing

Training step:
Ground Truth를 다음 스텝의 input으로

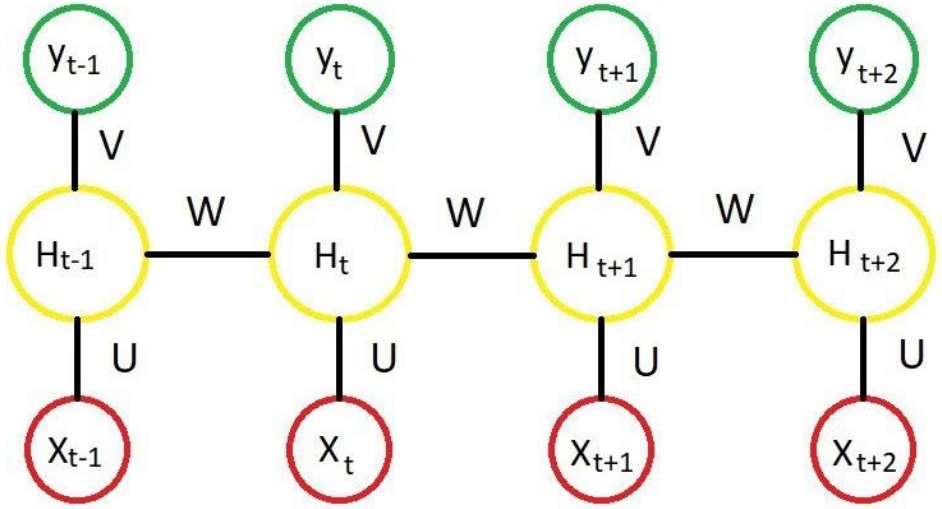
Dev, test step:
predicted output을 다음 스텝의 input으로

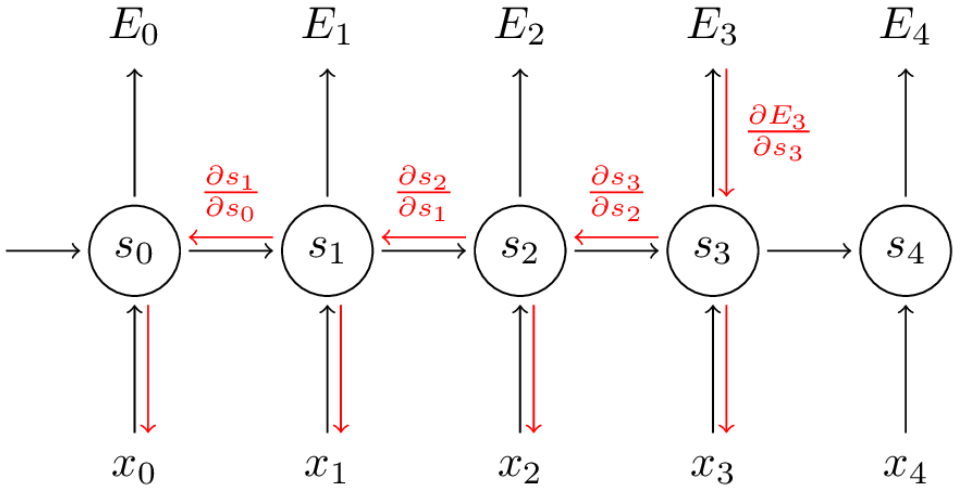
왜죠..?
Train 관점과 test 관점의 차이를 두고 해석해보기



같은 weight Matrix가 곱해지는 현상..

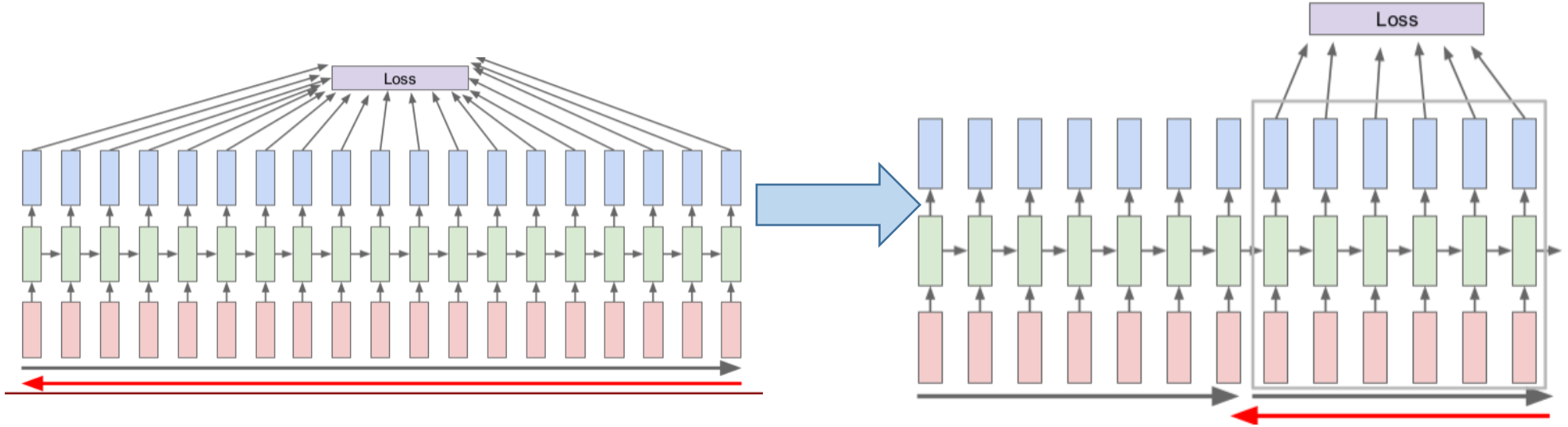
Gradient Vanishing + 다른 뉴럴넷에서는 없었던 Gradient Exploding 문제 발생
(time step이 길어질수록)





Solution 1.

BPTT vs TPTT

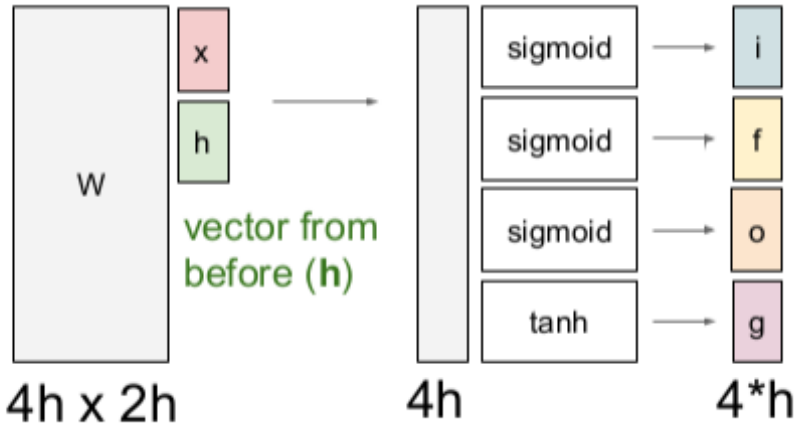


<Batch 단위로 sequence를 잘라서 넣기!>

그러면 이전 batch의 정보는....?

Solution 2.

그러면 각 step마다 들어가는 weight를 다르게 만들 수는 없을까...?
Weight를 reasonable하게 조절하는 gate를 만들기! - LSTM, GRU



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

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

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

< LSTM >

그러면 각 step마다 들어가는 weight를 다르게 만들 수는 없을까...?

Weight를 reasonable하게 조절하는 gate를 만들기! - LSTM, GRU

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

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

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

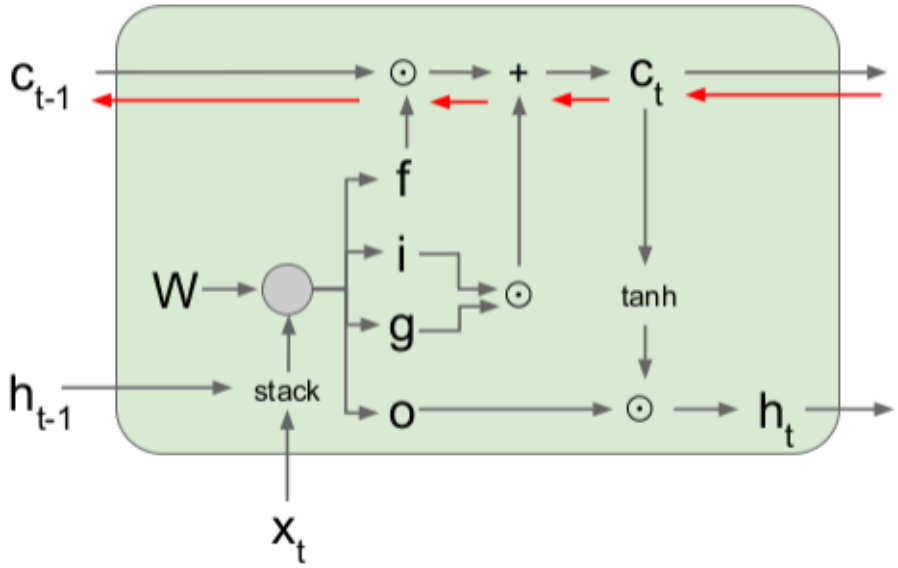
각 Gate에는 weight에 비선형 함수 Sigmoid를 사용하여 이전 step과 현재의 input을 얼마나 반영할지, 얼마나 잊어버릴지를 퍼센티지로 결정.

< 장점>
Cell에 들어가는 Gradient는 단 한번만 반영되어 gate에 weight를 뿌려주는 역할

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

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

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



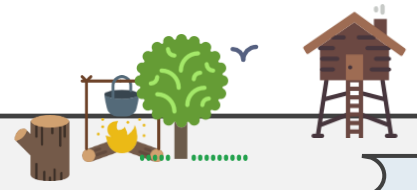
But.
RNN 구조의 아키텍쳐는 이러한 문제에 자유로울 수 없다. 계속해서 새로운 구조를 만들어 내는 중.. > Attention, Transformer 등등..

End

참고 자료

CS231n : http://cs231n.stanford.edu/syllabus.html

Deeplearning.ai : https://www.youtube.com/watchv=efWlOCE_6HY&list=PL1w8k37X_6L_s4ncq-swTBvKDWnRSrinI



BOAZ