아티스트를 위한 머신러닝 & 딥러닝

텐서플로를 활용한 딥러닝 #7

서울대학교 & V.DO / 김대식

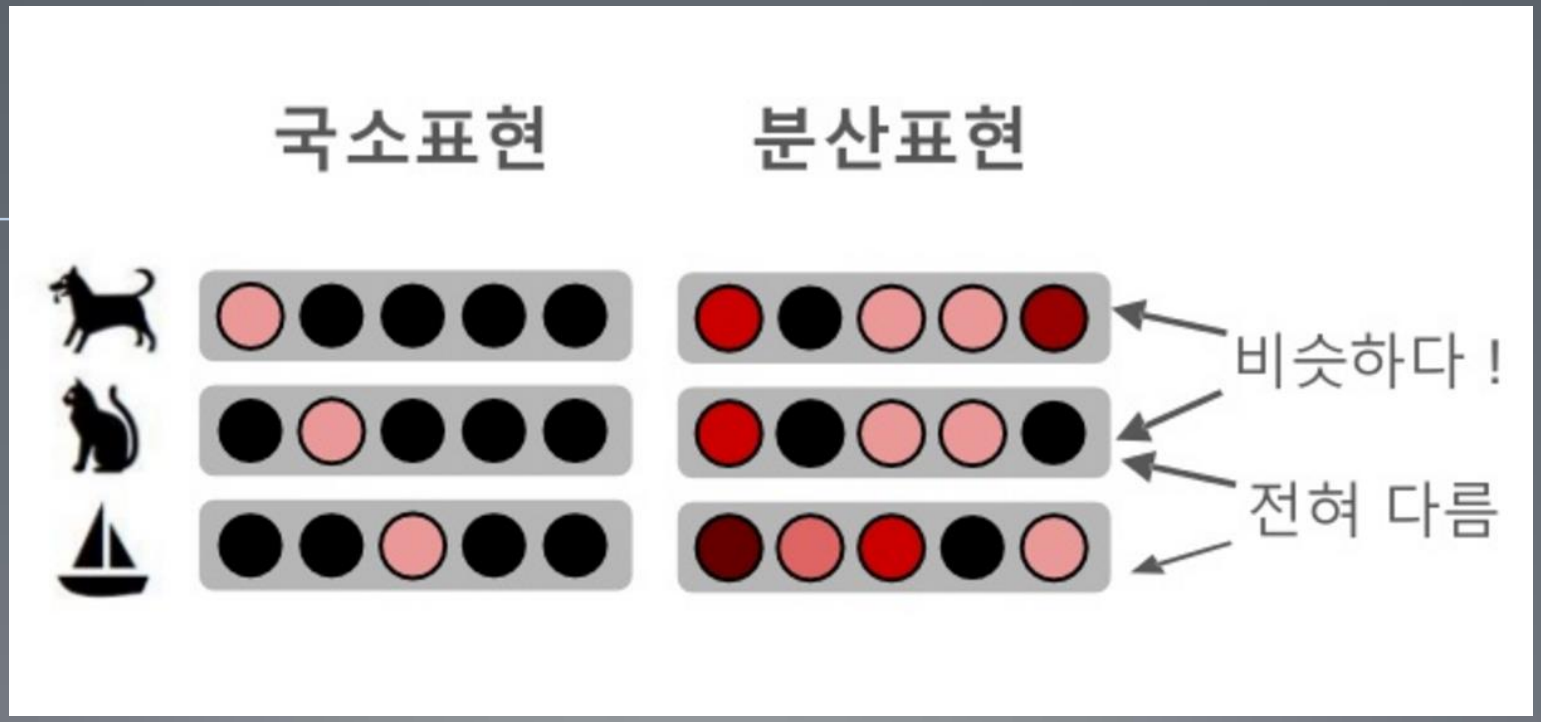
art center nabi v.do

Recap

Word Vector

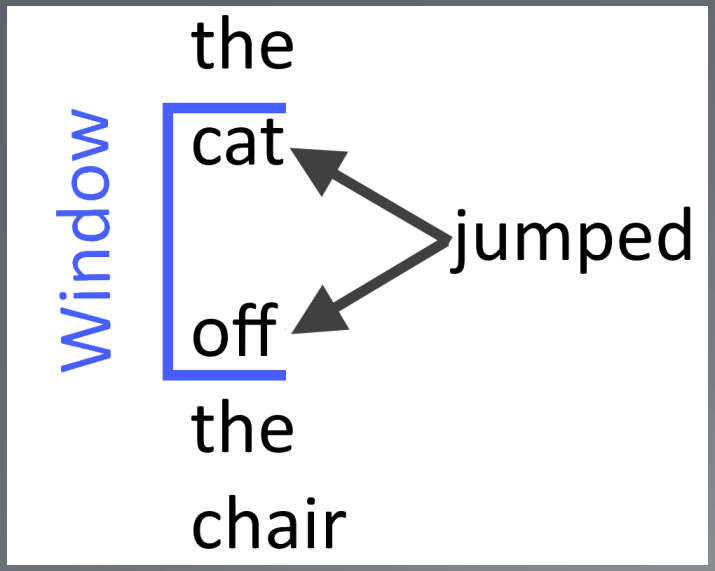
art center nabi v.do

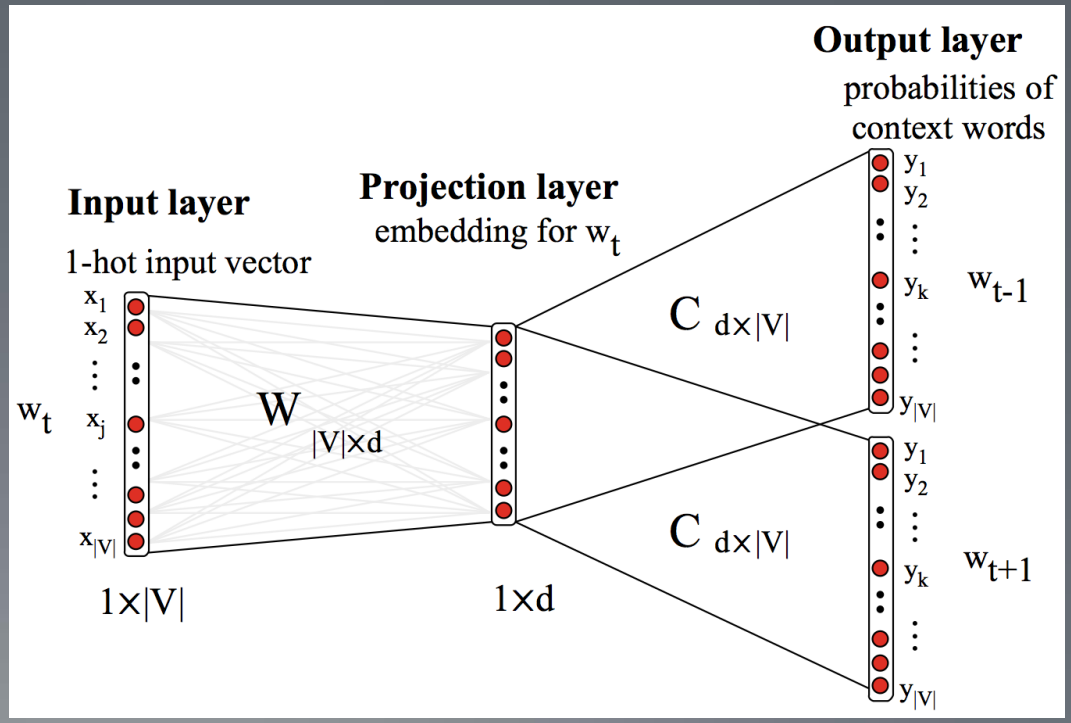
분산 표현 (Distributed Representation)



- CBOW
  - 문맥으로 부터 단어 예측
  - 소규모 데이터 셋에 성능 유리
- Skip-gram
  - 단어로부터 문맥 예측
  - 대규모 데이터셋에 유리

Skip-gram 구조


Window
the
cat
jumped
off
the
chair


Output layer
probabilities of
context words
Input layer
1-hot input vector
Projection layer
embedding for w_t
w_t
x_1 x_2 x_j $x_{|V|}$
W
|V|×d
1×|V|
1×d
C d×|V|
C d×|V|
y_1 y_2 y_k $y_{|V|}$
w_{t-1}
w_{t+1}

Word Embedding

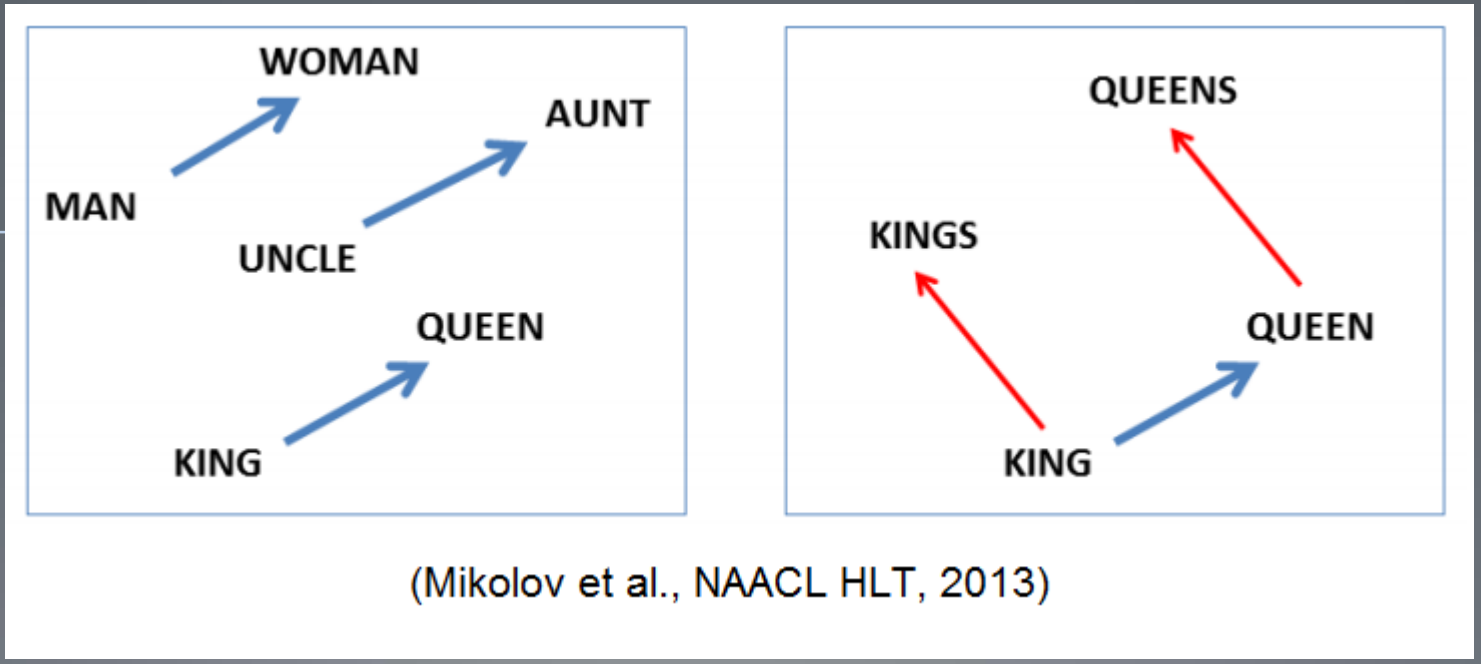
1-hot input

W

Distributed representation

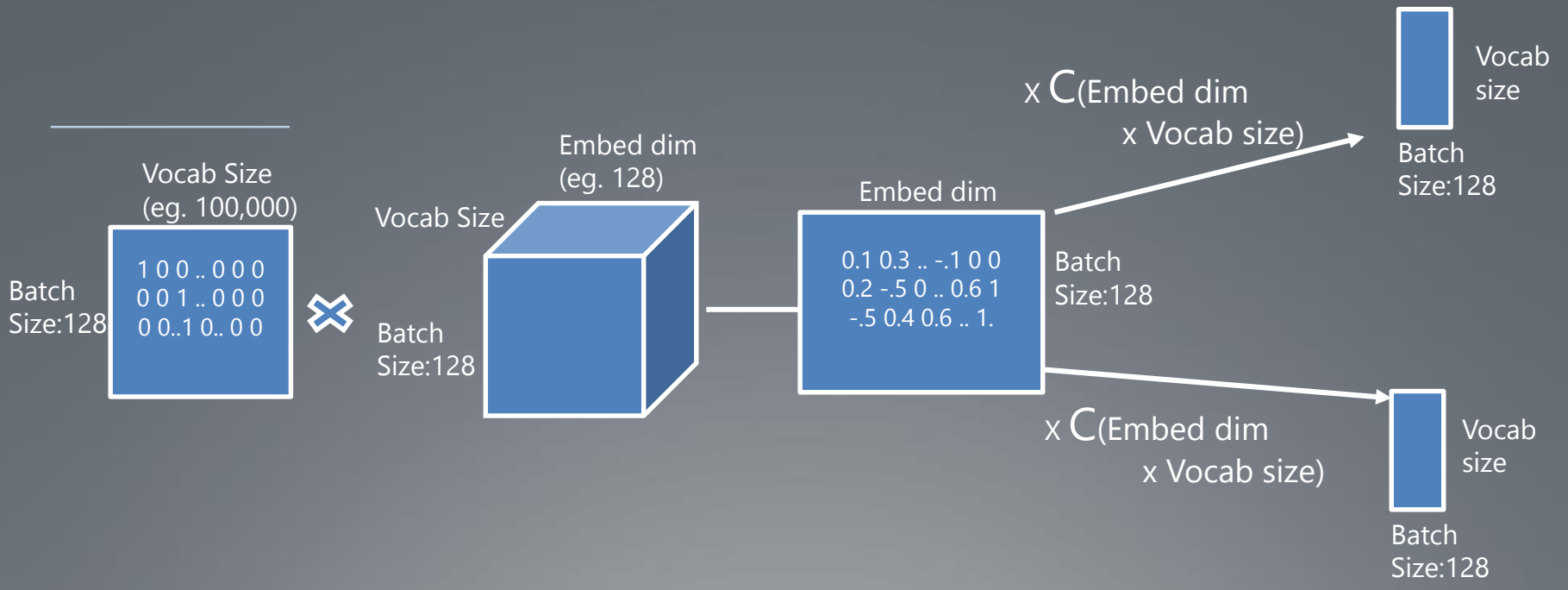
$$[0 \quad 0 \quad 0 \quad 1 \quad 0] \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} = [10 \quad 12 \quad 19]$$

Results



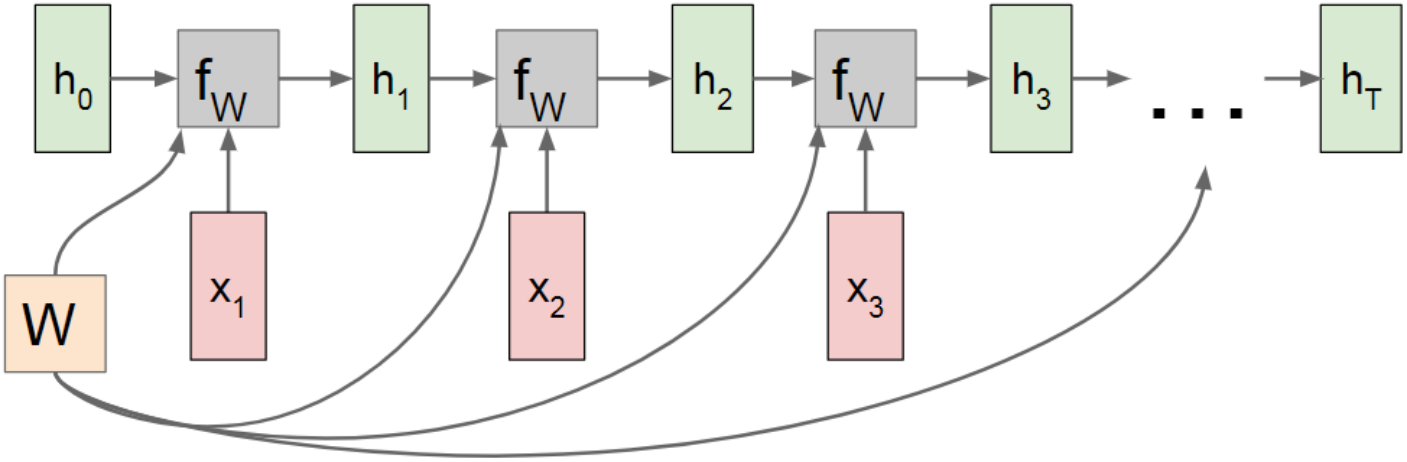
(Mikolov et al., NAACL HLT, 2013)

실습 : skip-gram 예제

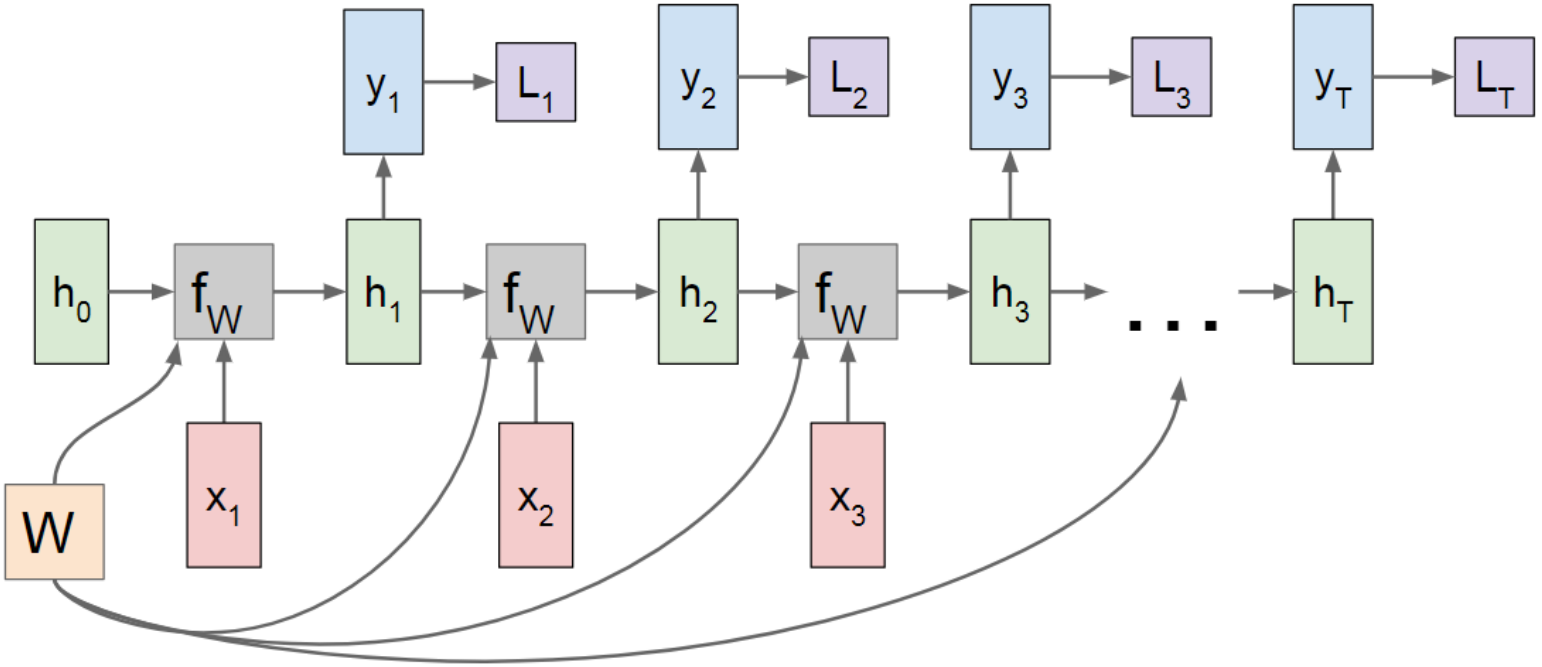


RNN: Computational Graph

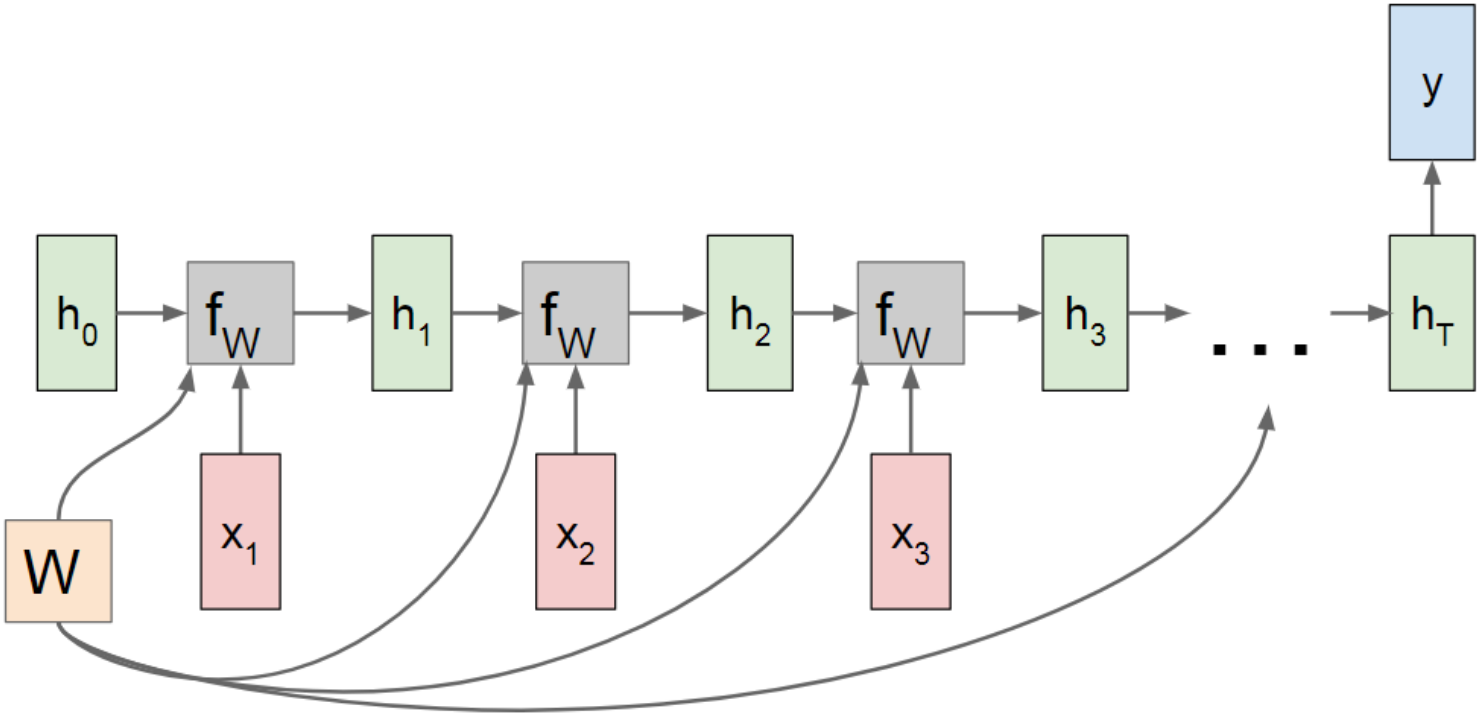
Re-use the same weight matrix at every time-step



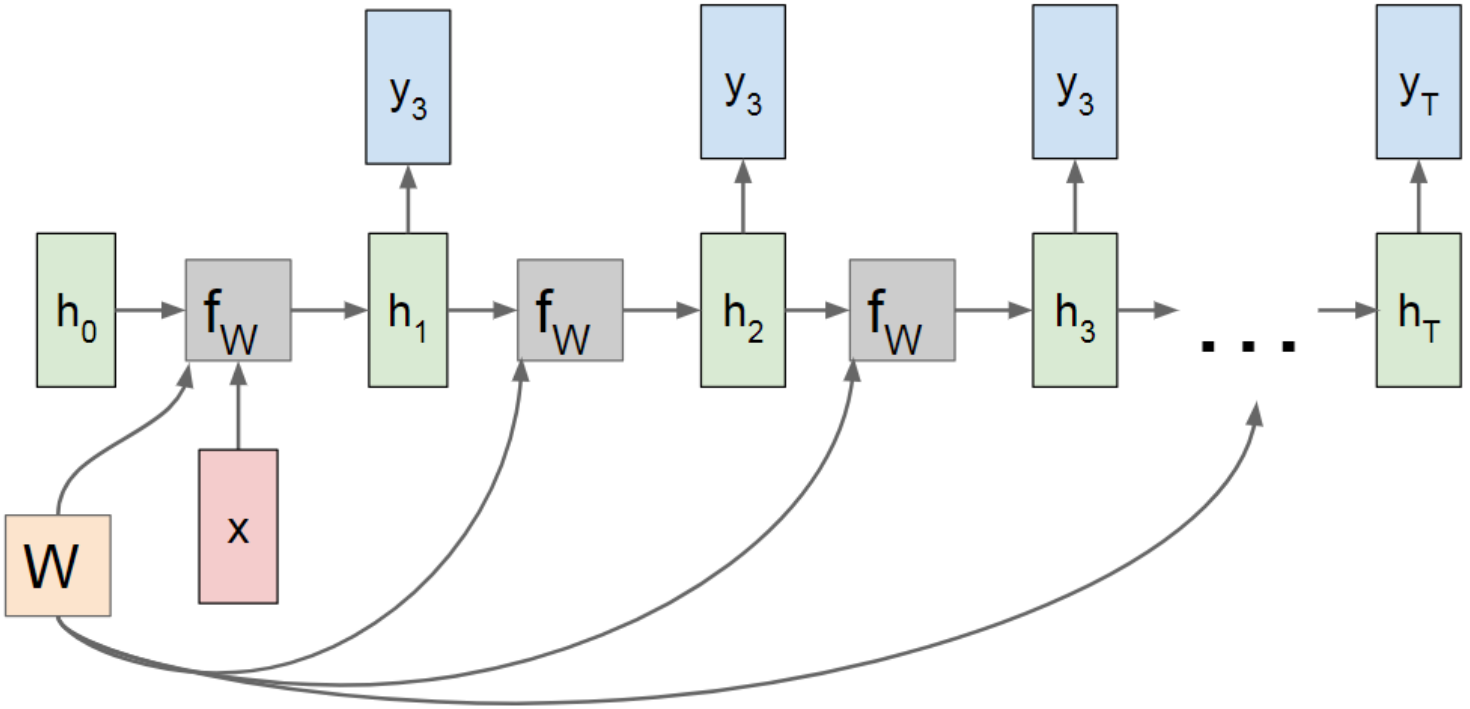
RNN: Computational Graph: Many to Many



RNN: Computational Graph: Many to One



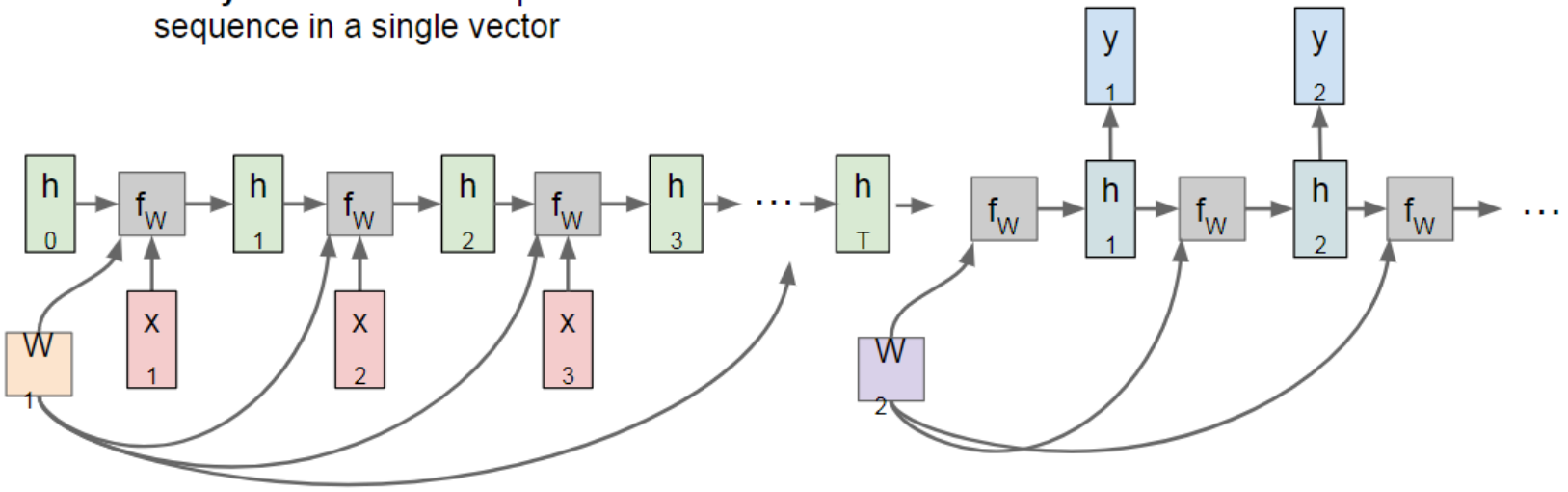
RNN: Computational Graph: One to Many



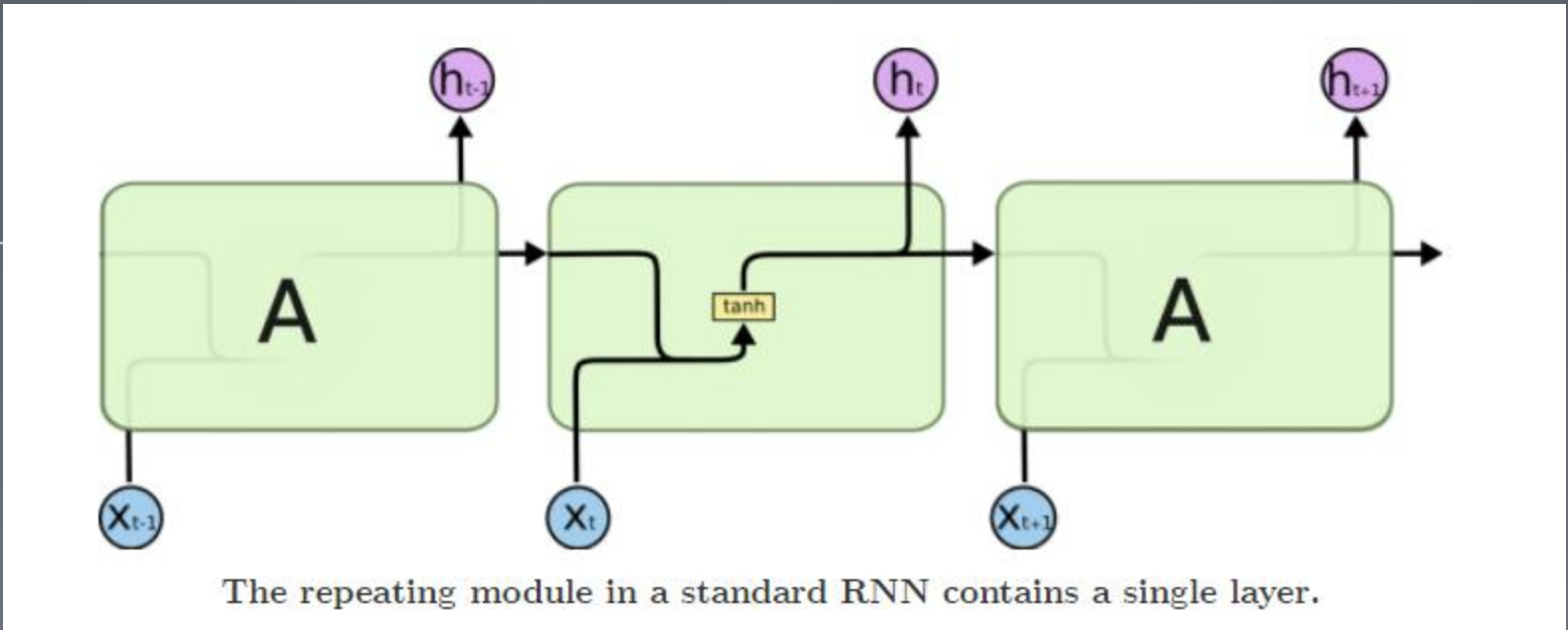
Sequence to Sequence: Many-to-one + one-to-many

Many to one: Encode input sequence in a single vector

One to many: Produce output sequence from single input vector

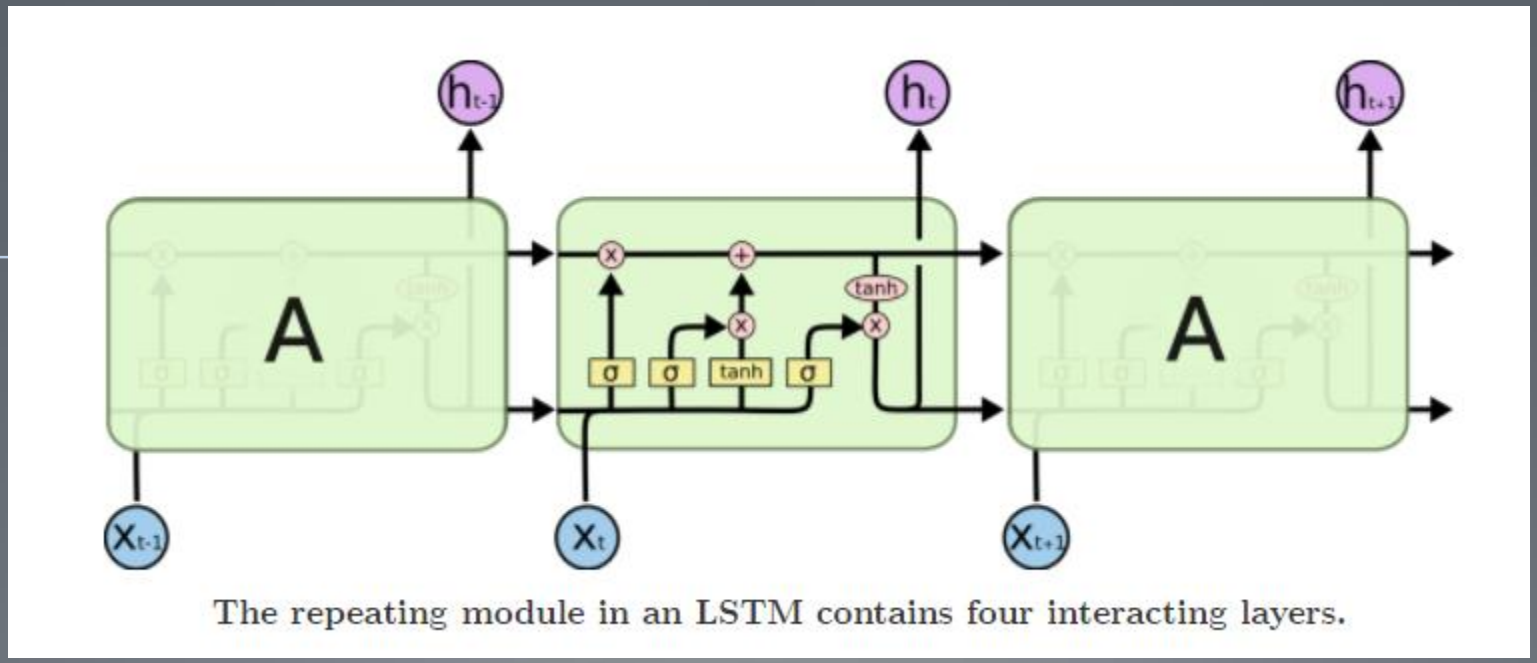


RNN Cell



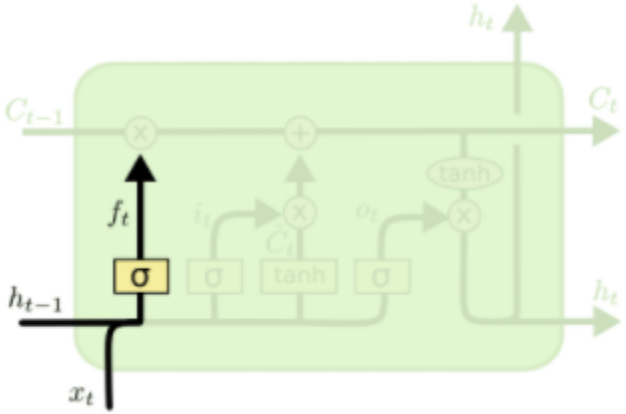
The repeating module in a standard RNN contains a single layer.

LSTM Cell



The repeating module in an LSTM contains four interacting layers.

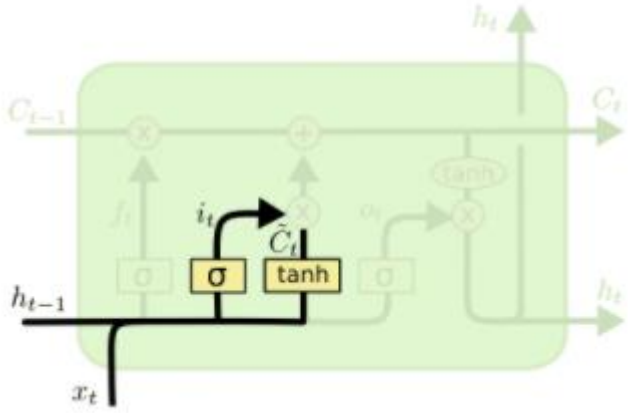
LSTM Cell



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

art center nabi v.do

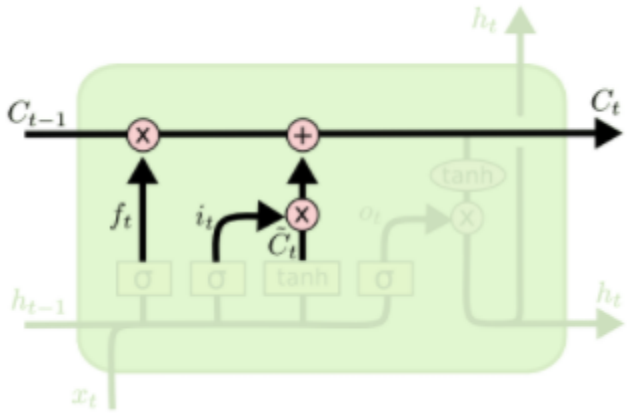
LSTM Cell



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

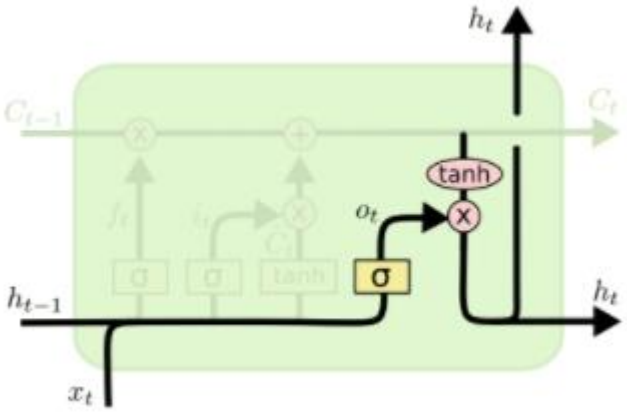
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

LSTM Cell



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

LSTM Cell



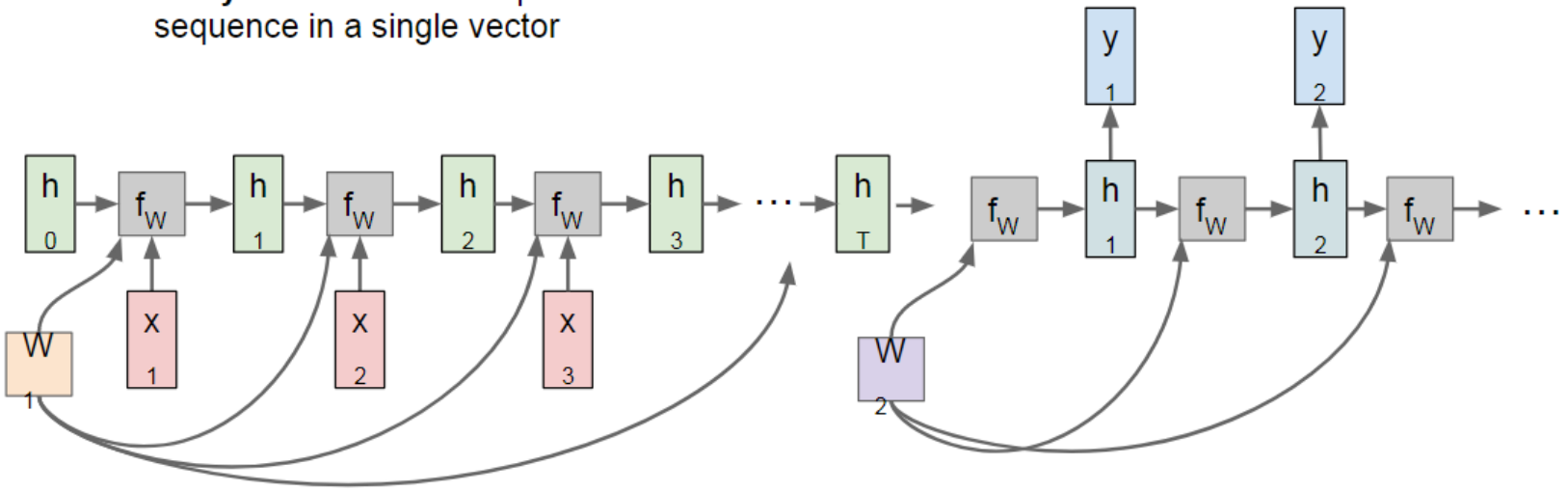
$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

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

Many to one: Encode input sequence in a single vector

One to many: Produce output sequence from single input vector

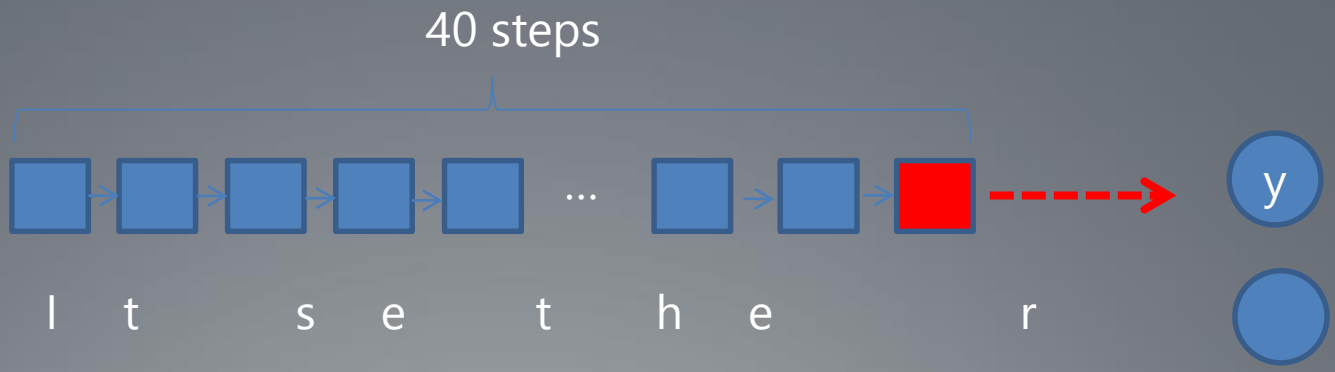


실습: Text Generation(1/5)

- RNN이 가장 많이 이용되는 분야인 NLP 예제
- 그 중 word단위가 아닌 character단위로 텍스트 분석 및 예측
- 알파벳 character 전후 관계와 문장 전체의 information을 이용

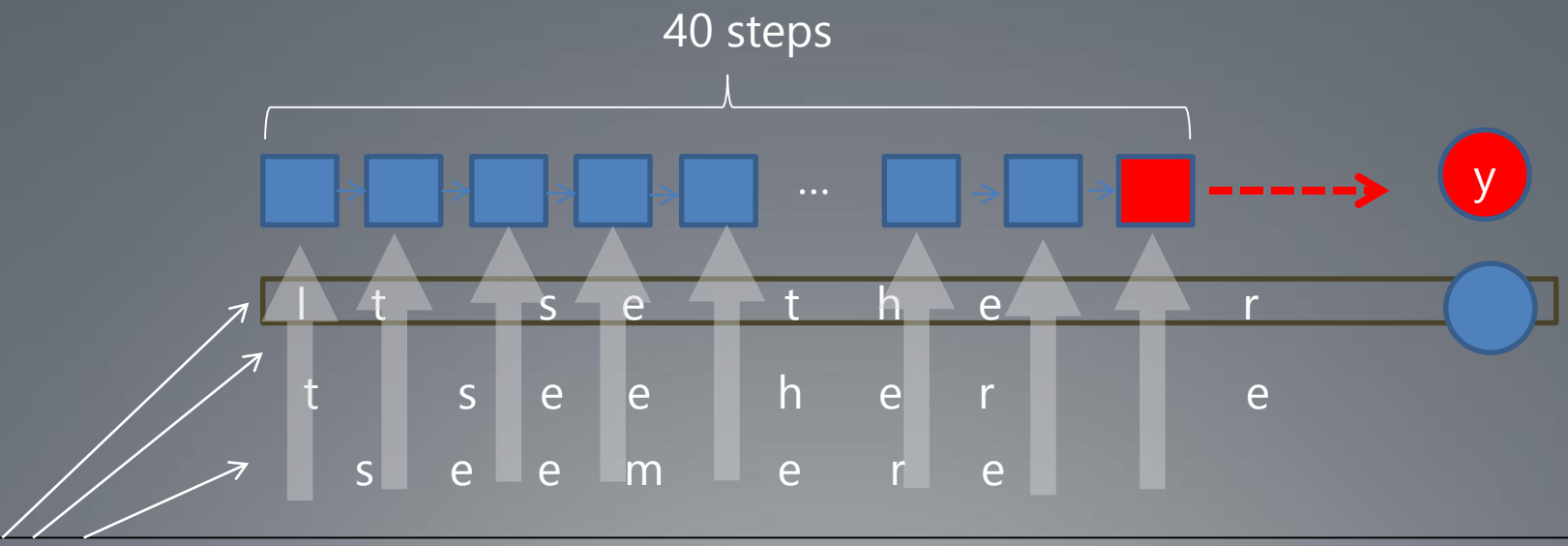
Text Generation (2/5)

- Dataset : nietzsche의 선악의 저편 text (OSIA/data)
- 40 steps의 LSTM 1 layer를 이용하여 다음 character 예측



Text Generation (3/5)

- 데이터 전처리
  - Character 59개를 모두 index화
  - 1 character씩 움직이면서 sentence와 예측할 다음 character를 target으로



Text Generation (4/5)

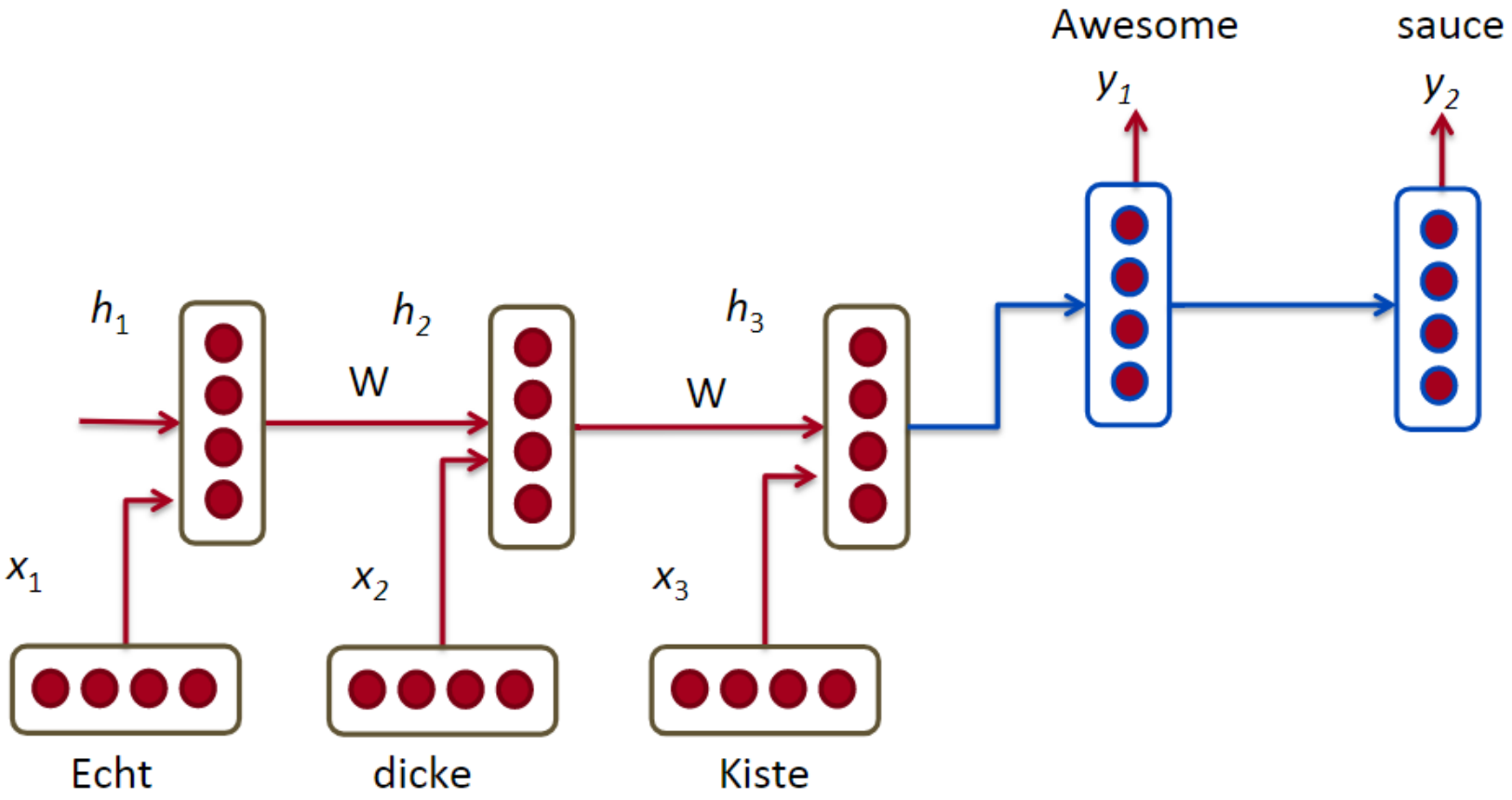
- Training Detail
  - Adam optimizer 사용
  - Learning rate = 0.01
  - Batch size = 128
  - LSTM Hidden cell의 수 = 128

Text Generation (5/5)

- Character Sampling
  - 1000 iteration마다 200 characters 연속 생성
  - 생성되는 character를 다시 input으로 사용
  - 트레이닝이 진행될수록 문장을 이루는 character 생성

RNN Translation Model Extensions

1. Train different RNN weights for encoding and decoding

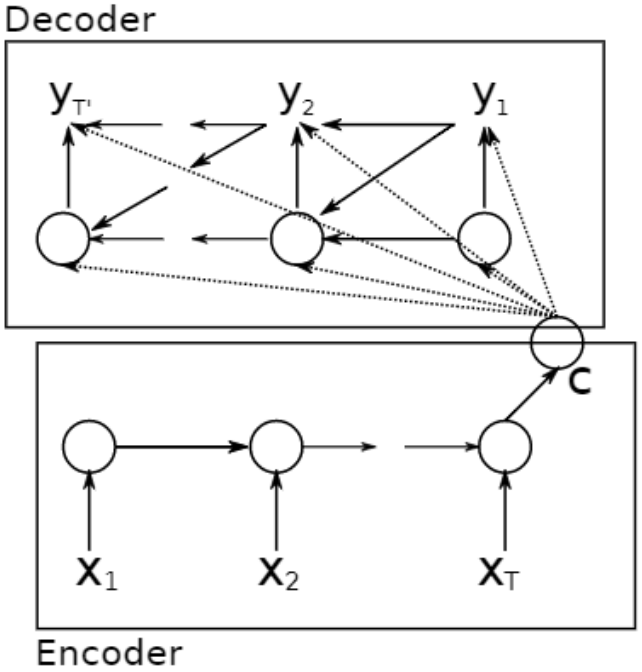


RNN Translation Model Extensions

Notation: Each input of ϕ has its own linear transformation matrix. Simple: $h_t = \phi(h_{t-1}) = f\left(W^{(hh)} h_{t-1}\right)$

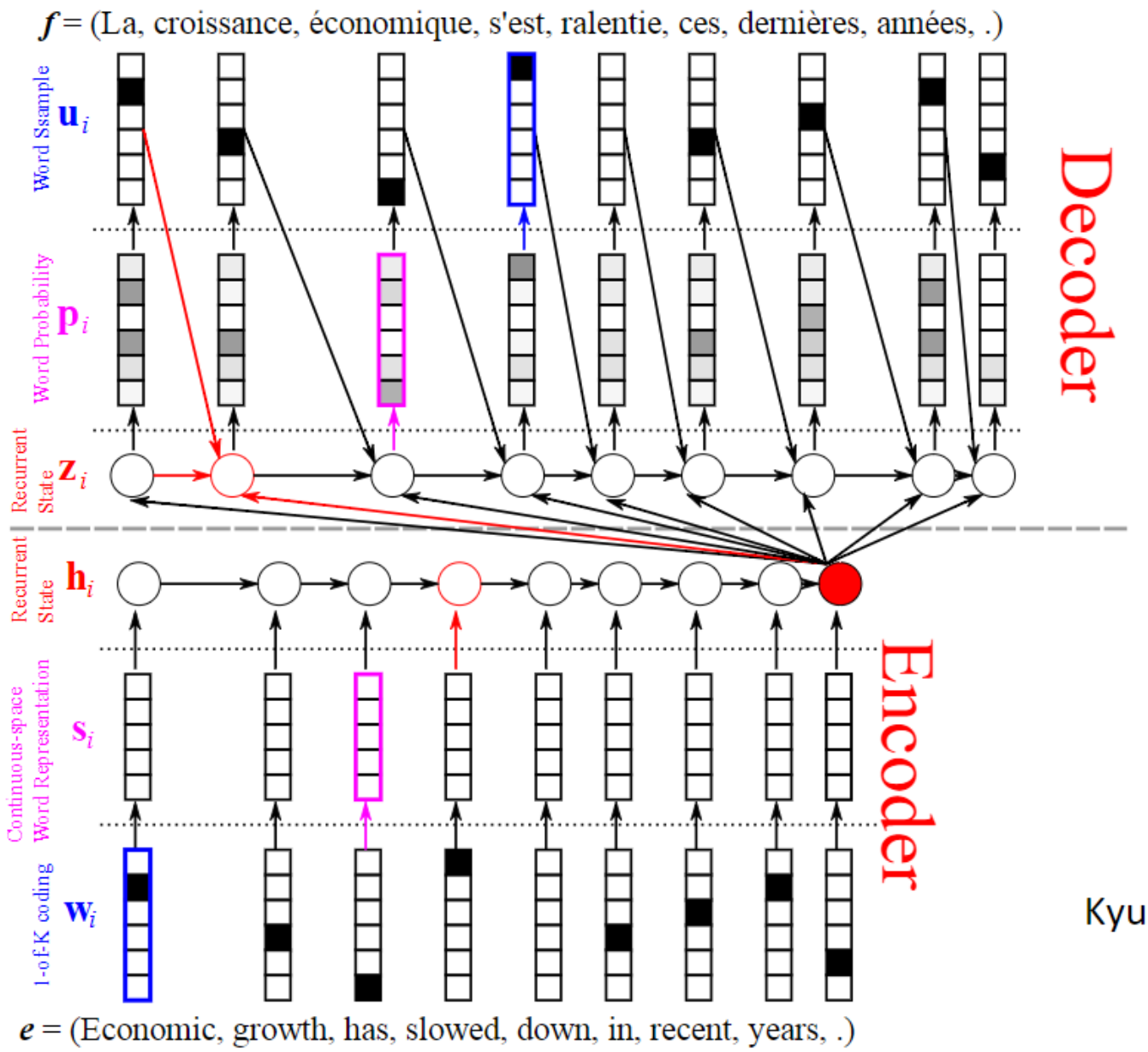
2. Compute every hidden state in decoder from
   - Previous hidden state (standard)
   - Last hidden vector of encoder c=h_T
   - Previous predicted output word y_{t-1}

$$h_{D,t} = \phi_D(h_{t-1}, c, y_{t-1})$$



Cho et al. 2014

Different picture, same idea

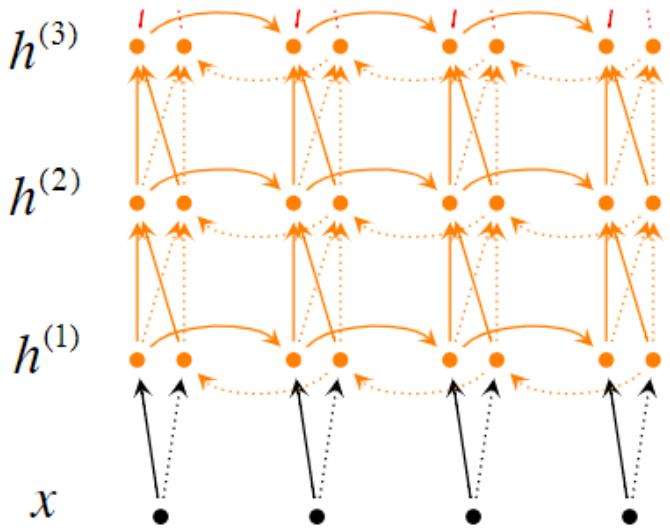


Kyunghyun Cho et al. 2014

RNN Translation Model Extensions

3. Train stacked/deep RNNs with multiple layers

4. Potentially train bidirectional encoder

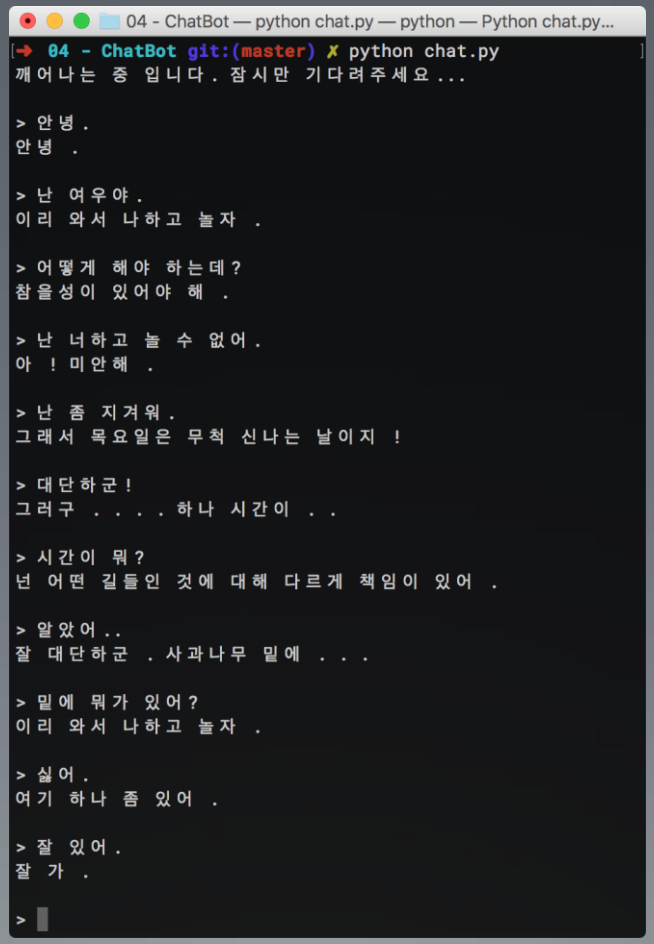


5. Train input sequence in reverse order for simple optimization problem: Instead of A B C → X Y, train with C B A → X Y

RNN 활용예제

art center nabi v.do

챗봇



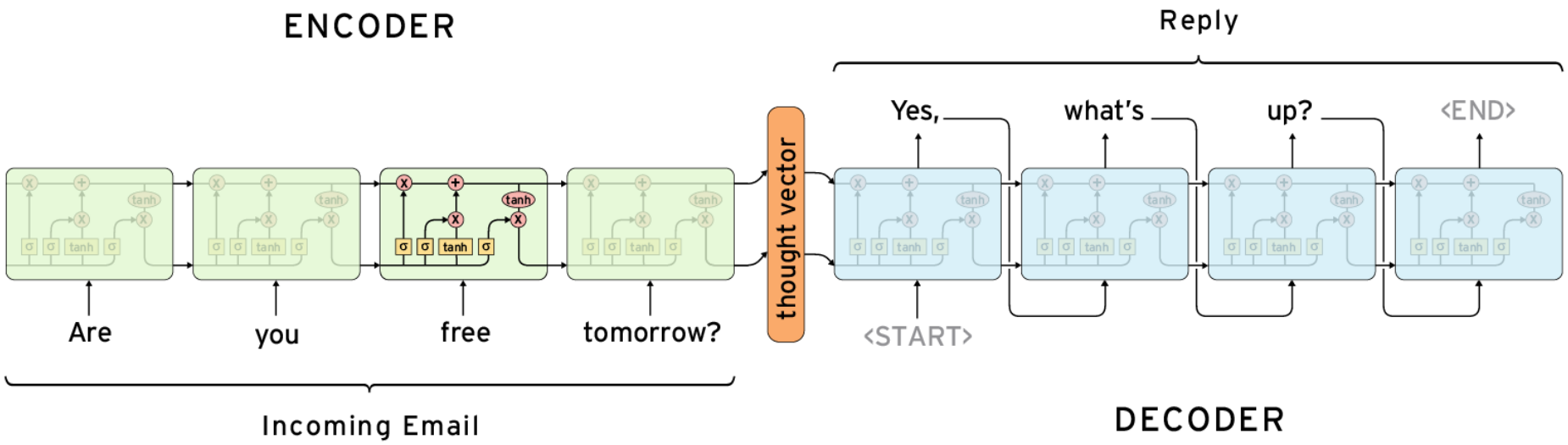
https://github.com/golbin/TensorFlow-Tutorials

art center nabi v.do

챗봇

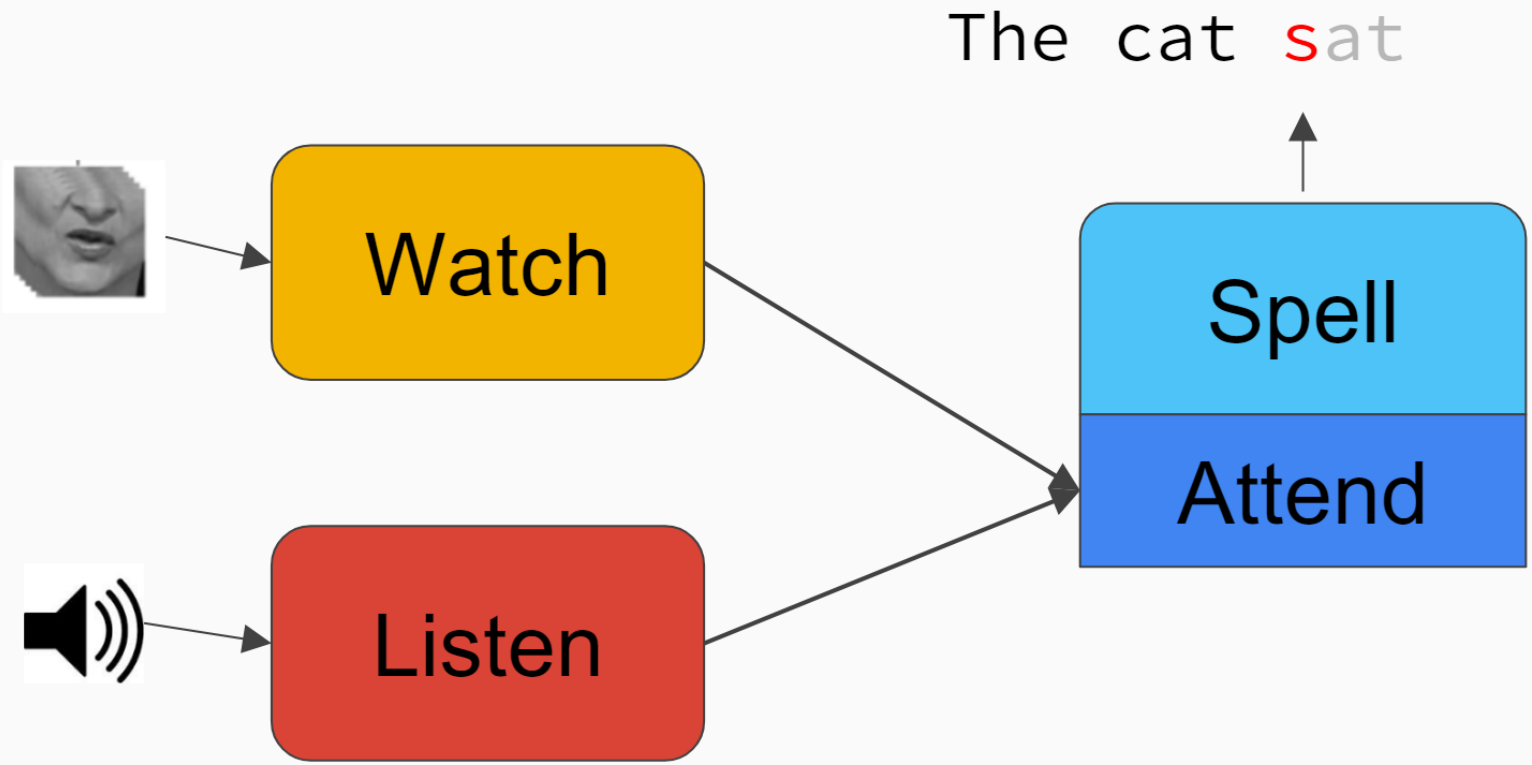
- Seq2Seq 모델을 사용
  - 질문과 답의 관계를 학습
  - 자연어에 강함
- 한계
  - 단순한 매핑, 논리적 사고 결여
  - 통계적 챗봇보다 더 나쁜경우 발생

챗봇



art center nabi v.do

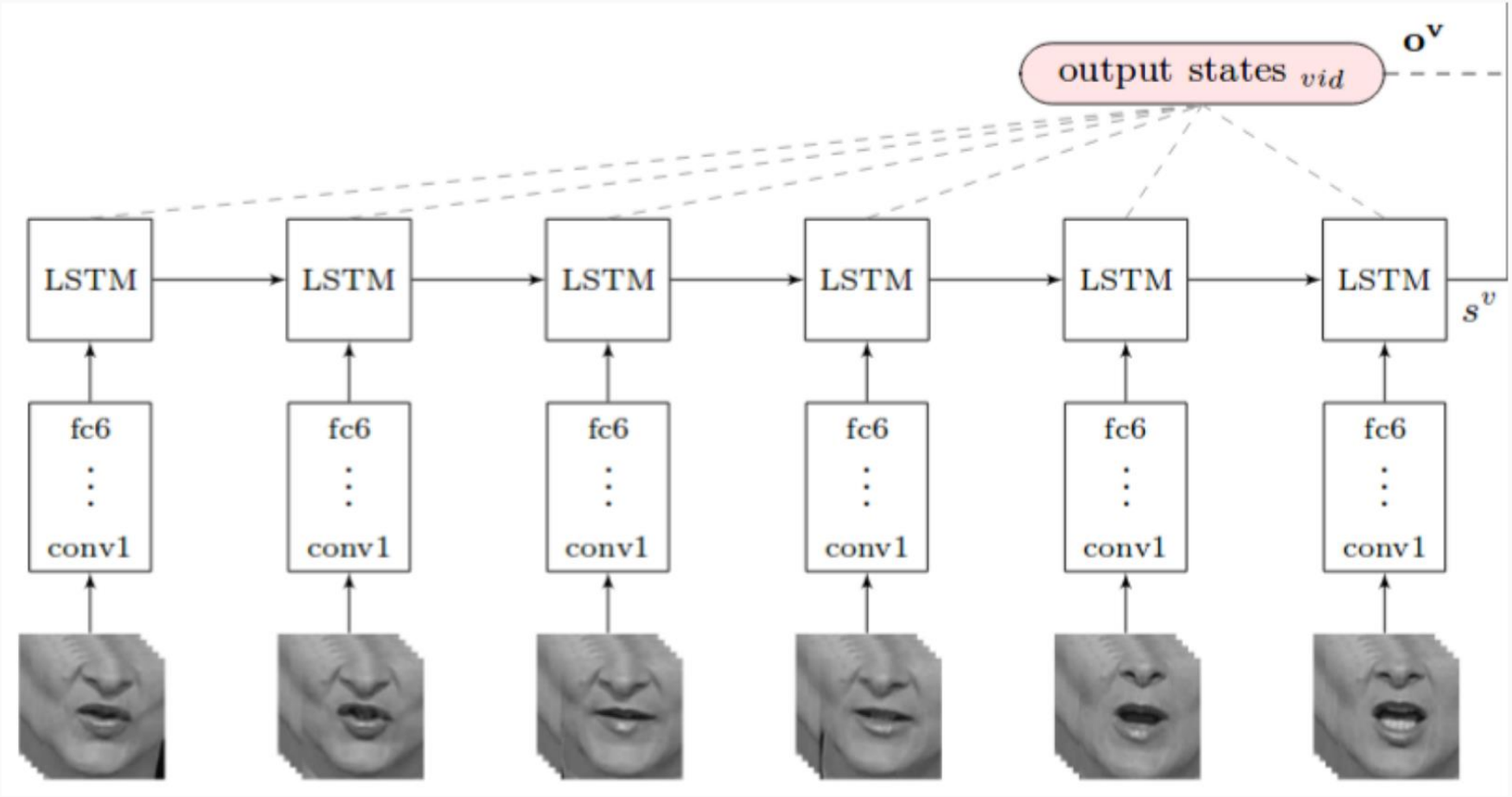
Lip reading



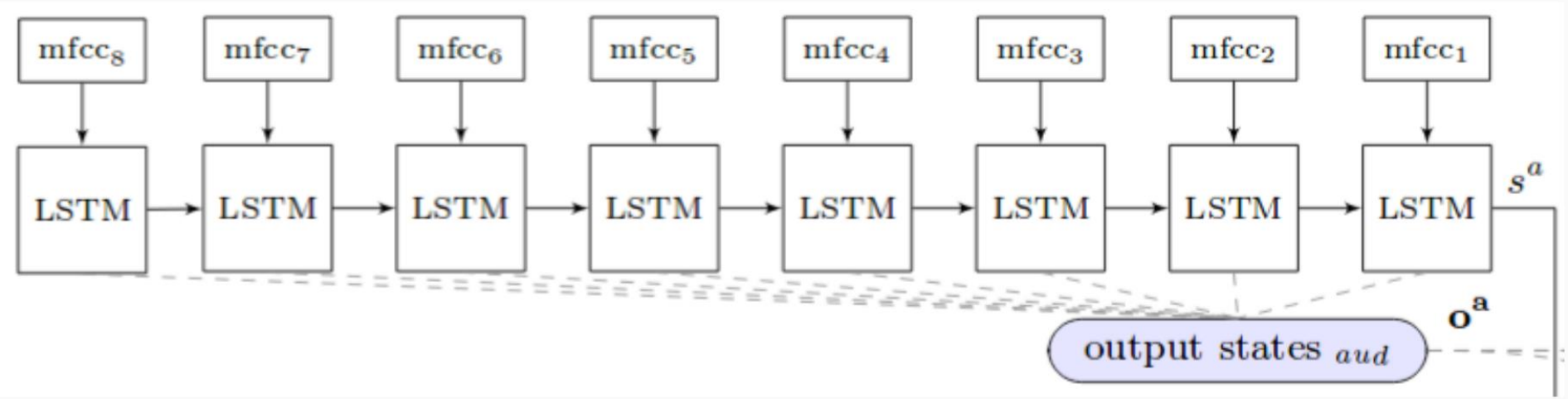
Lip reading

- 음성인식과 입술 인식의 결합
  - 서로의 한계점을 보완
  - RNN을 이용한 모델
- 링크
  - https://www.youtube.com/watch?v=5aogzAUPilE

Watch

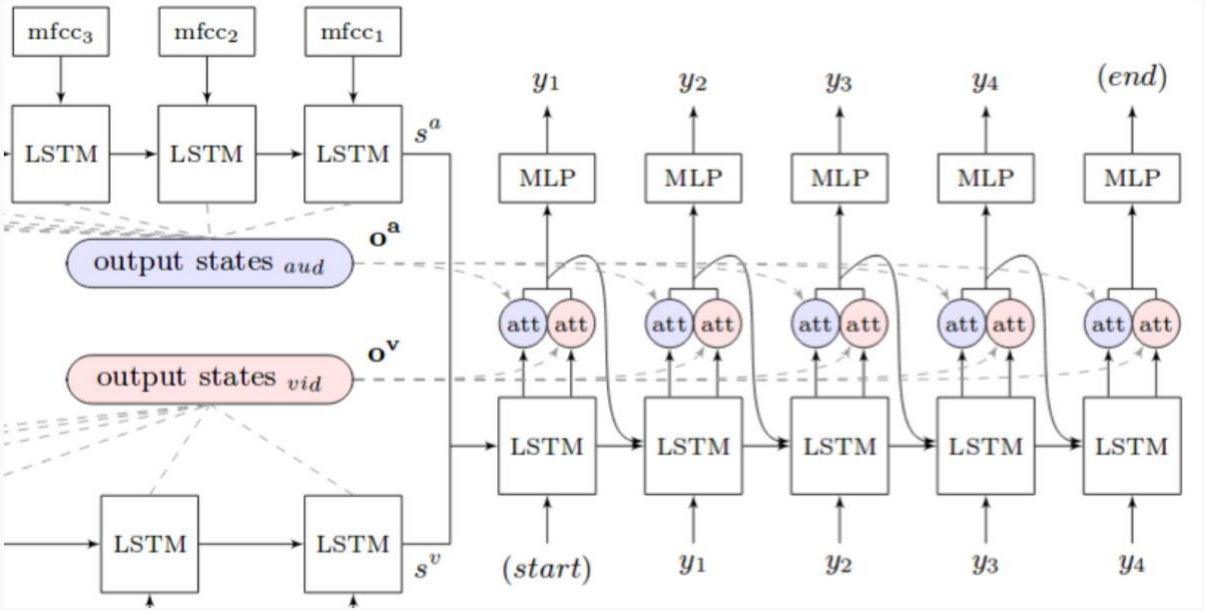

output states vid
o^v
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
s^v
fc6
conv1
fc6
conv1
fc6
conv1
fc6
conv1
fc6
conv1
fc6
conv1

Listen


mfcc8
mfcc7
mfcc6
mfcc5
mfcc4
mfcc3
mfcc2
mfcc1
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
s^a
$\mathbf{o^a}$
output states aud

Attend and Spell

$$
\begin{aligned}
h_k^d, o_k^d &= \texttt{LSTM}(h_{k-1}^d, y_{k-1}, c_{k-1}^v, c_{k-1}^a) \\
c_k^v &= \mathbf{o}^v \cdot \texttt{Attention}^{\texttt{v}}(h_k^d, \mathbf{o}^v) \\
c_k^a &= \mathbf{o}^a \cdot \texttt{Attention}^{\texttt{a}}(h_k^d, \mathbf{o}^a) \\
P(y_i|\mathbf{x}^v, \mathbf{x}^a, y_{<i}) &= \texttt{softmax}(\texttt{MLP}(o_k^d, c_k^v, c_k^a))
\end{aligned}
$$



Dataset

Channel	Series name	# hours	# sent.
BBC 1 HD	News†	1,584	50,493
BBC 1 HD	Breakfast	1,997	29,862
BBC 1 HD	Newsnight	590	17,004
BBC 2 HD	World News	194	3,504
BBC 2 HD	Question Time	323	11,695
BBC 4 HD	World Today	272	5,558
All		4,960	118,116

마지막 시간

- 못다한 주제들
  - Attention
  - Object Detection
  - Deep speech
- Discussion or QA

감사합니다