Neural Language Models

Knowledge and Language Engineering Lab

POSTECH
POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY

목차

- 신경망 언어 모델 소개
- LSTMs 기반 언어 모델 실습

신경망 언어모델 소개

언어 모델

- 언어 모델?
  - 문장 또는 단어열에 대한 확률 분포
  - m개의 단어열이 주어졌을 때 m개의 단어열이 나타날 확률을 계산
  - P(I am a boy) = 0.7
  - P(I a am boy) = 0.02
- 적용 예
  - 품사 태깅
    - $P(I_{noun}\ am_{verb}\ a_{article}\ boy_{noun})$=?
  - 기계 번역
    - P(high winds tonight) > P(large winds tonight)
  - 철자 교정
    - P(about fifteen minutes from) > P(about fifteen minuets from)
  - 기타 등등…

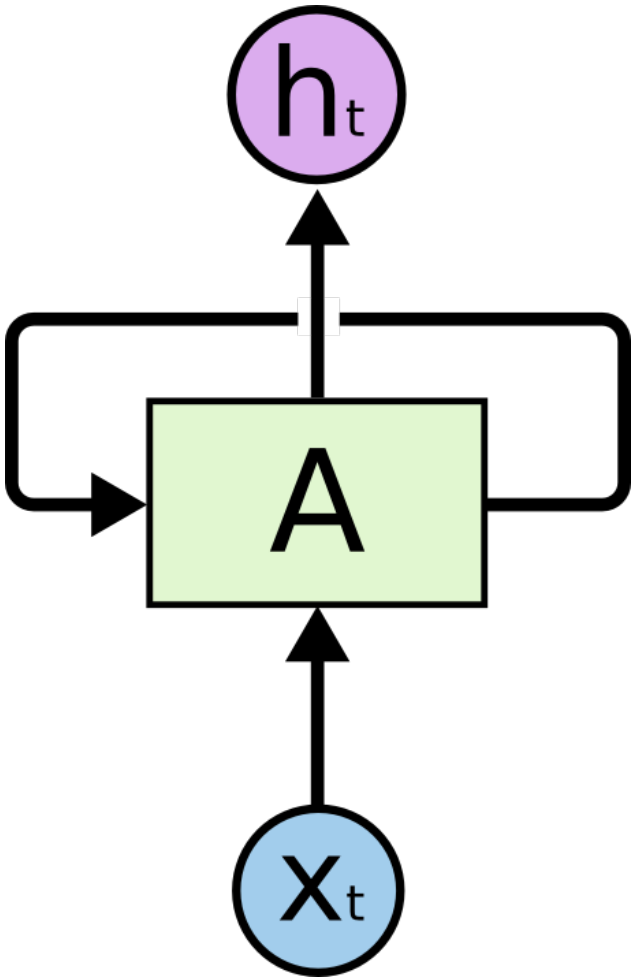
언어 모델

- 통계기반 언어 모델?
  - N-gram 언어 모델

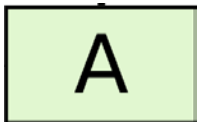
- 신경망 기반 언어 모델?
  - Recurrent neural network 기반 언어 모델

Recurrent Neural Networks

- 순환신경망 (Recurrent Neural Networks; RNNs)



Input



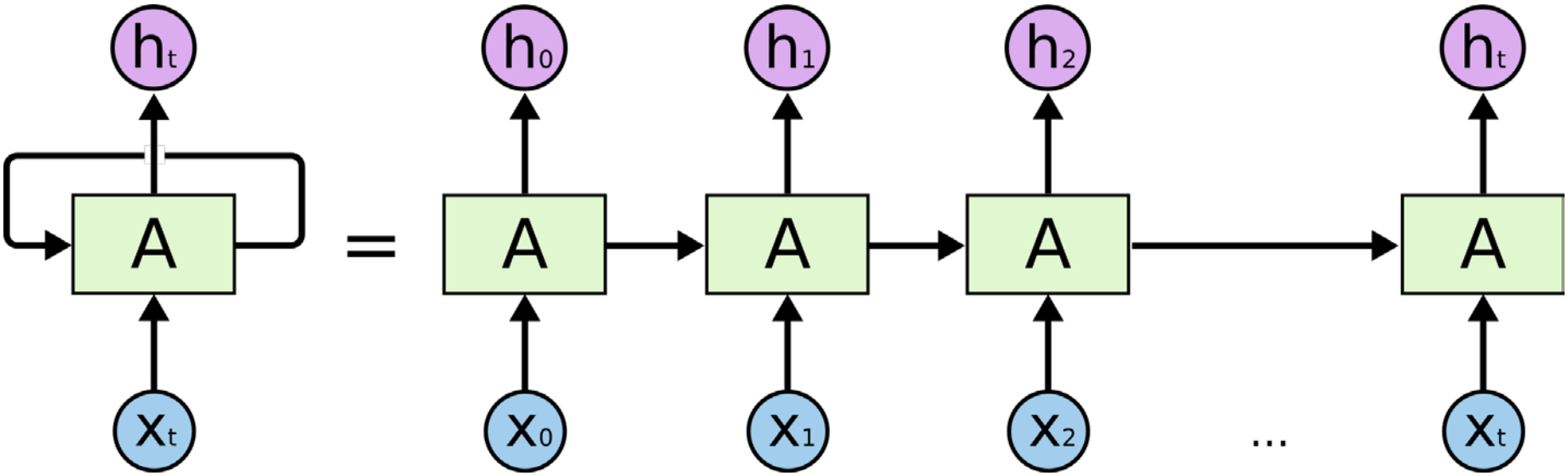
Cell

Output

t Timestep

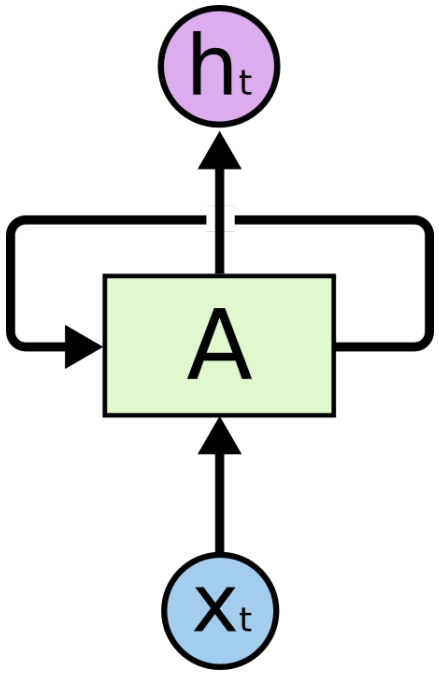
Recurrent Neural Networks

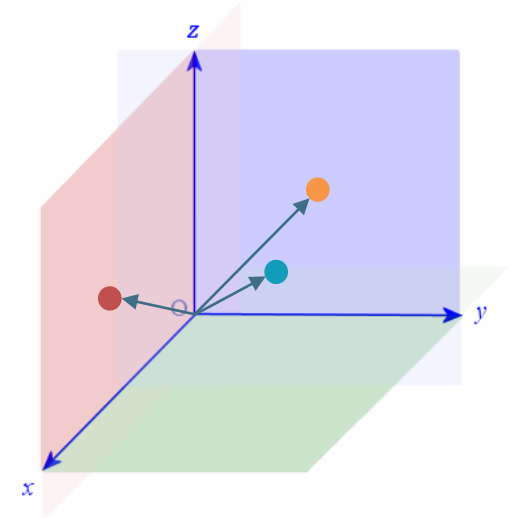
- 순환신경망 (Recurrent Neural Networks; RNNs)



Recurrent Neural Networks

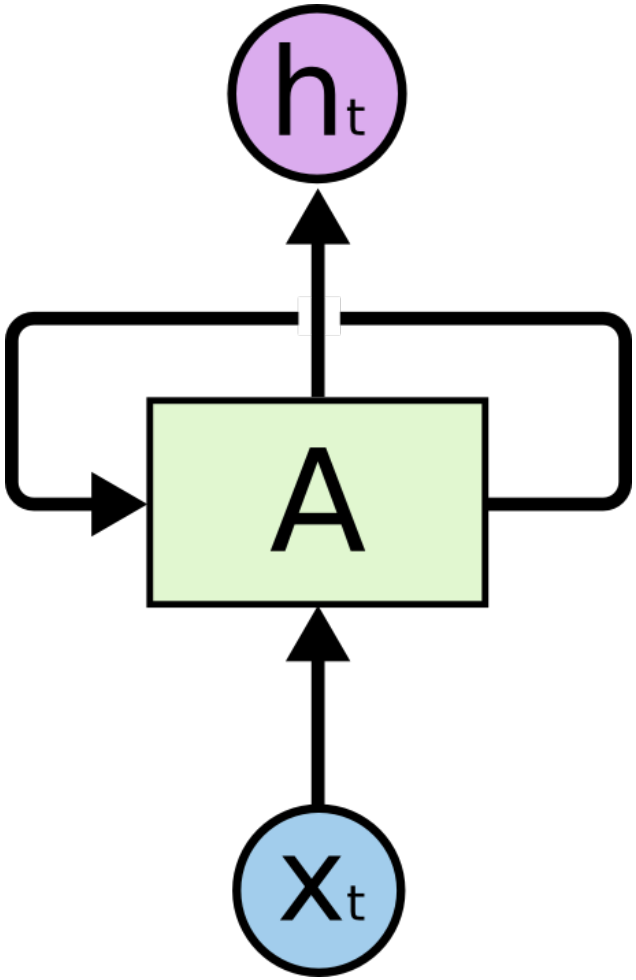
- 순환신경망 (Recurrent Neural Networks; RNNs)
  - 무작위 길이의 열 → 고정된 길이의 벡터 표현
    - I am a boy
    - Sometimes to understand a word's…
    - At your dictionary we try to gib…

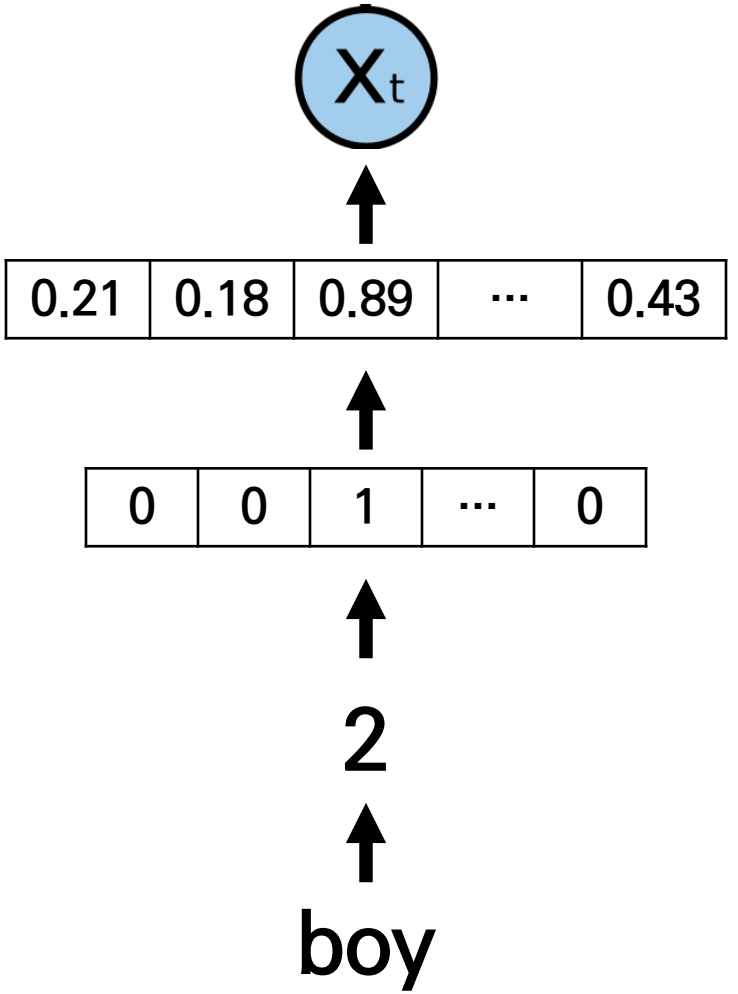




Recurrent Neural Networks

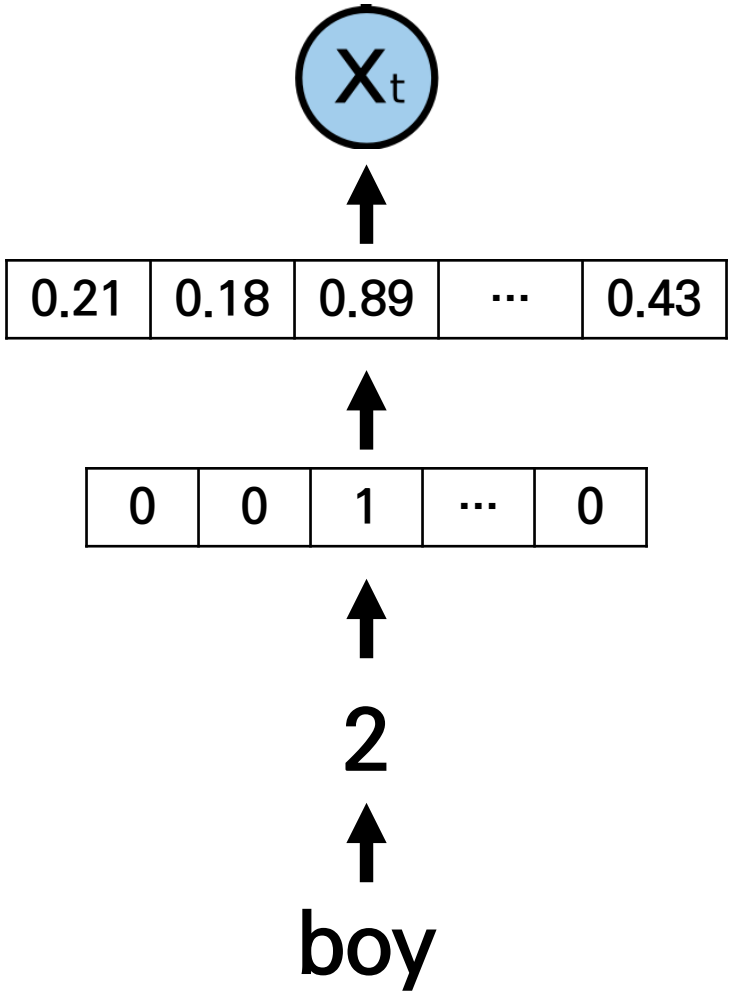
- RNN의 입력





Recurrent Neural Networks

- RNN의 입력: Embedding Layer (Word to Vector)



word index = 3

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

one-hot representation of word

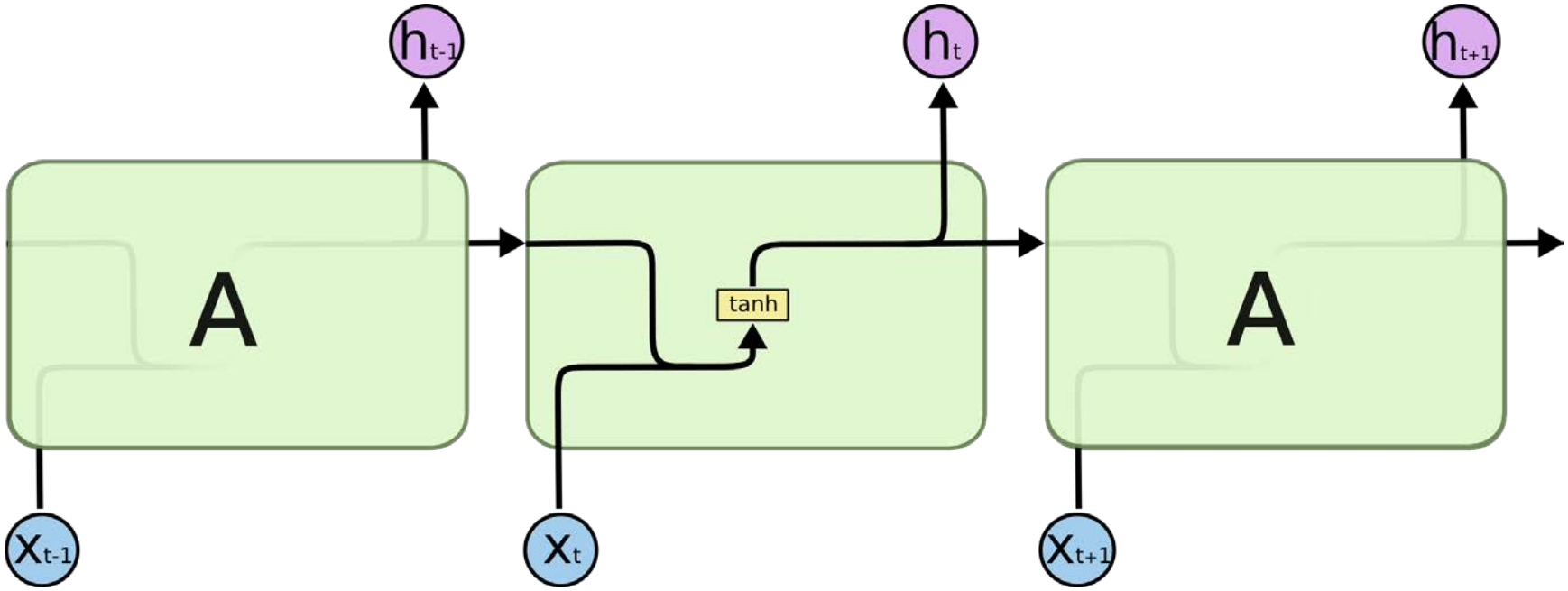
Word Embedding matrix
(Weight matrix)

$$= [10 \quad 12 \quad 19]$$

get embedding vector w.r.t word index

Recurrent Neural Networks

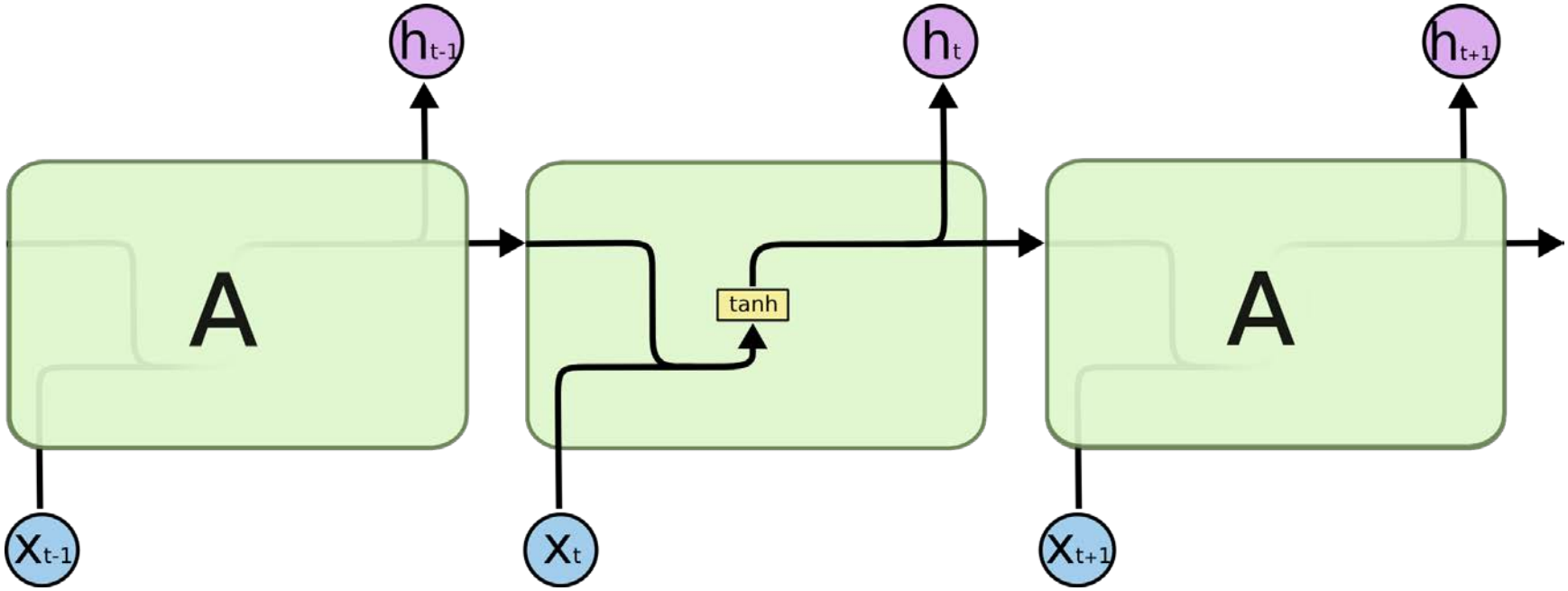
- RNN 출력



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

Recurrent Neural Networks

- Timestep마다 다른 Weight? Or weight sharing?



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

Recurrent Neural Networks

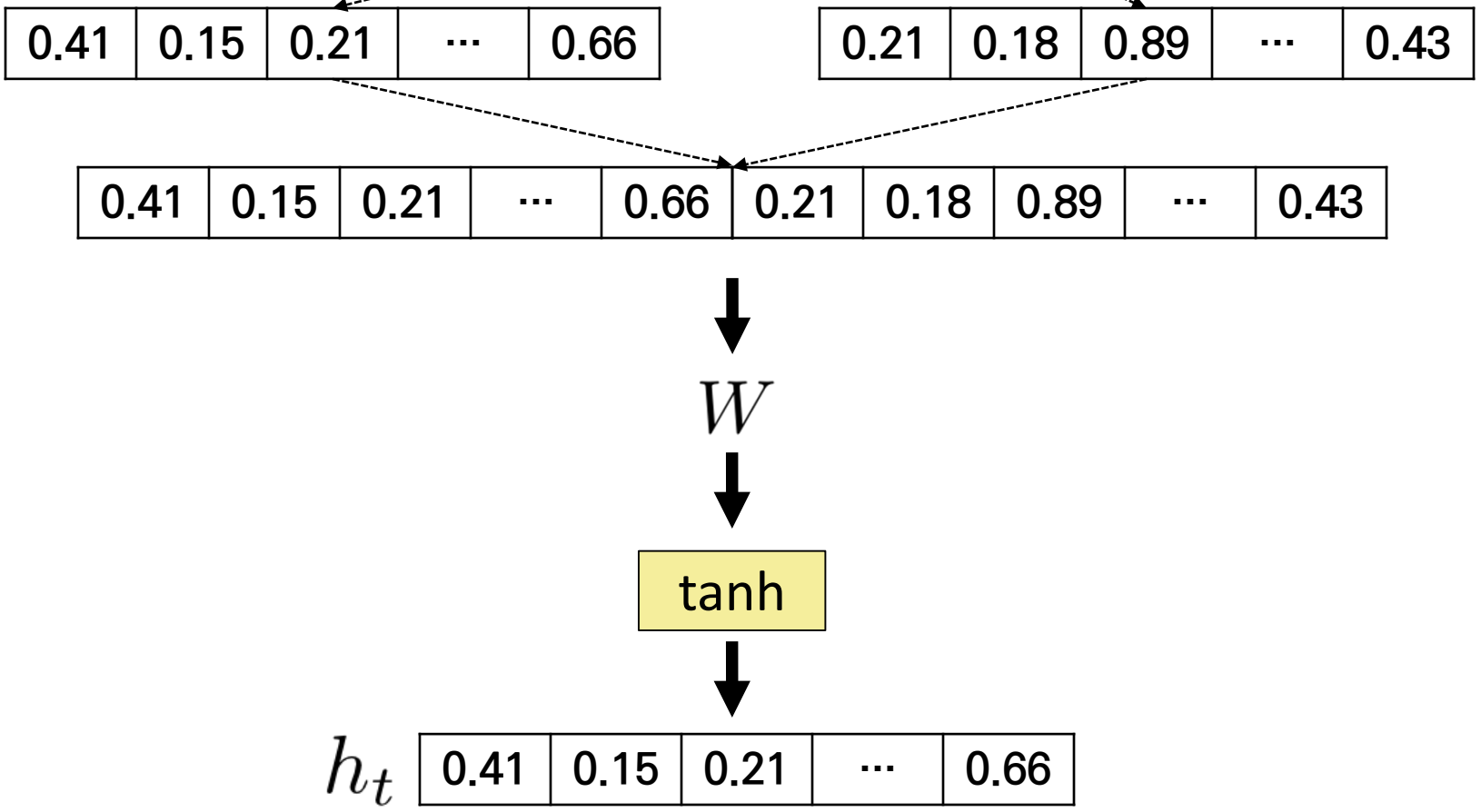
- Timestep마다 다른 weight 사용
  - 학습 파라미터의 수가 선형적으로 증가
  - 네트워크가 학습하지 못한 입력열에 대한 일반화 불가능
  - on monday it was snowing
  - it was snowing on Monday

- Weight sharing
  - 학습 파라미터 수 효율적
  - 학습 데이터 오버피팅 감소
  - 가변길이의 입력열을 가지는 모델링에 도움

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

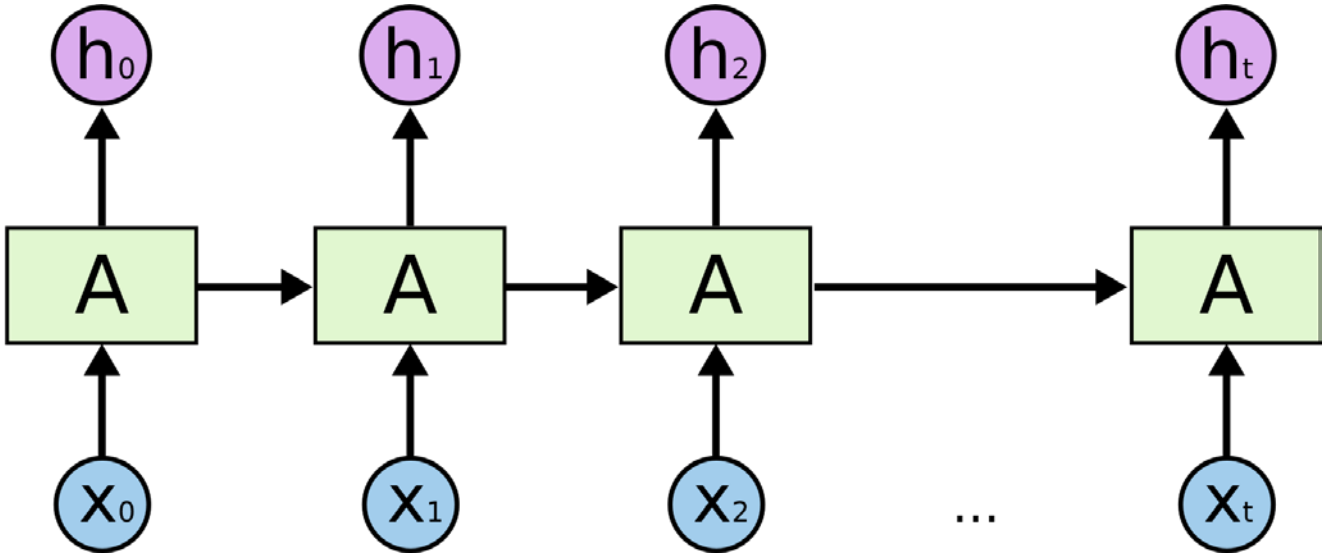
Recurrent Neural Networks

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



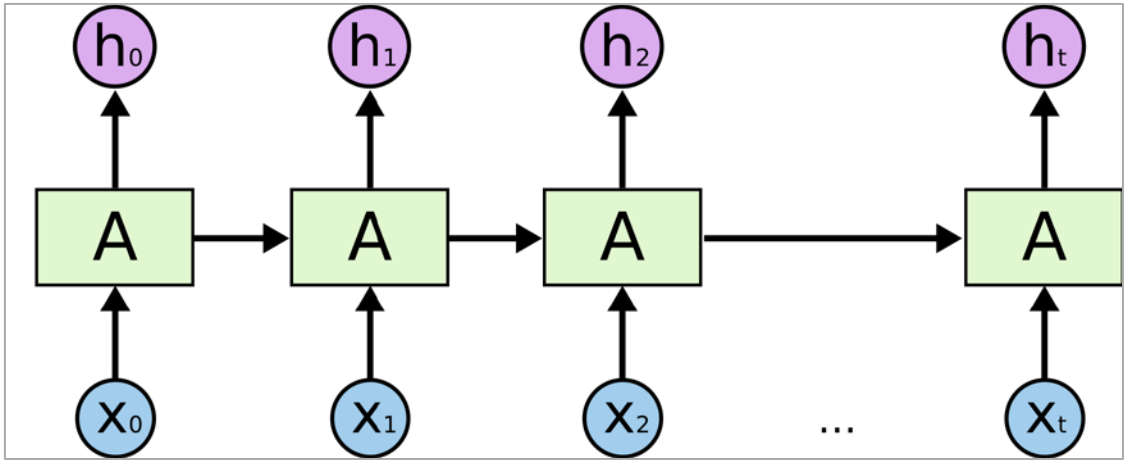
Recurrent Neural Networks

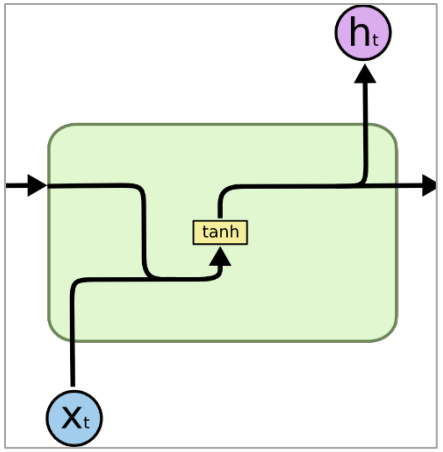
- 불행하게도, 길이가 긴 열 학습 어려움
  - Vanishing gradient problem



Recurrent Neural Networks

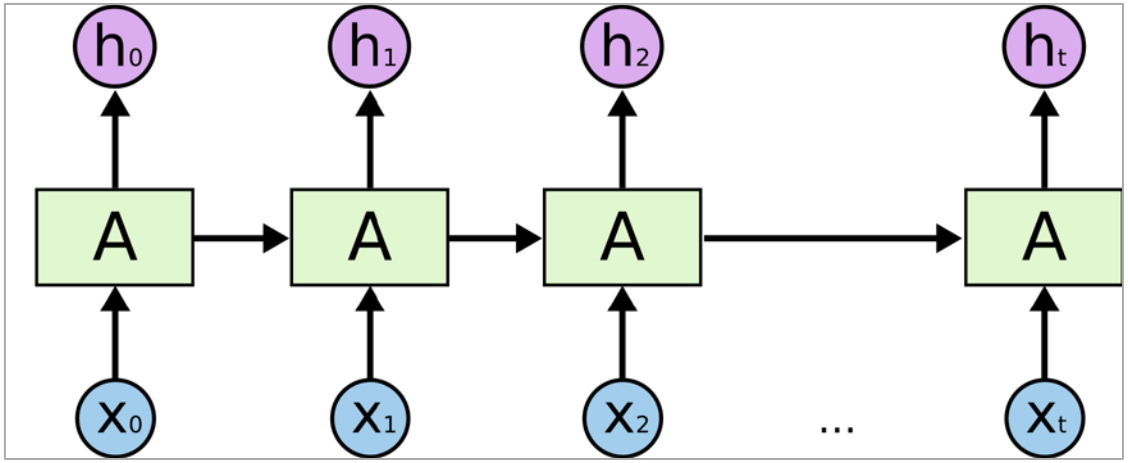
- 불행하게도, 길이가 긴 열 학습 어려움
  - Vanishing gradient problem

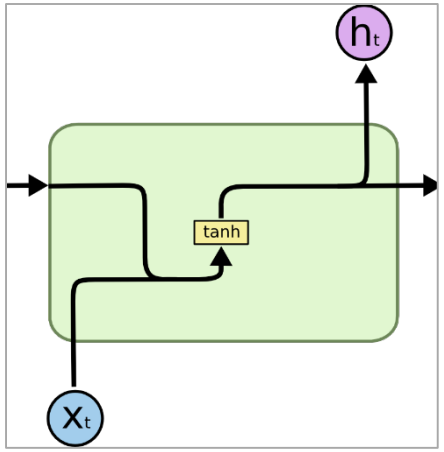




Recurrent Neural Networks

- 불행하게도, 길이가 긴 열 학습 어려움
  - Vanishing gradient problem
  - 장기 의존성 학습 어려움 (long-term dependency)



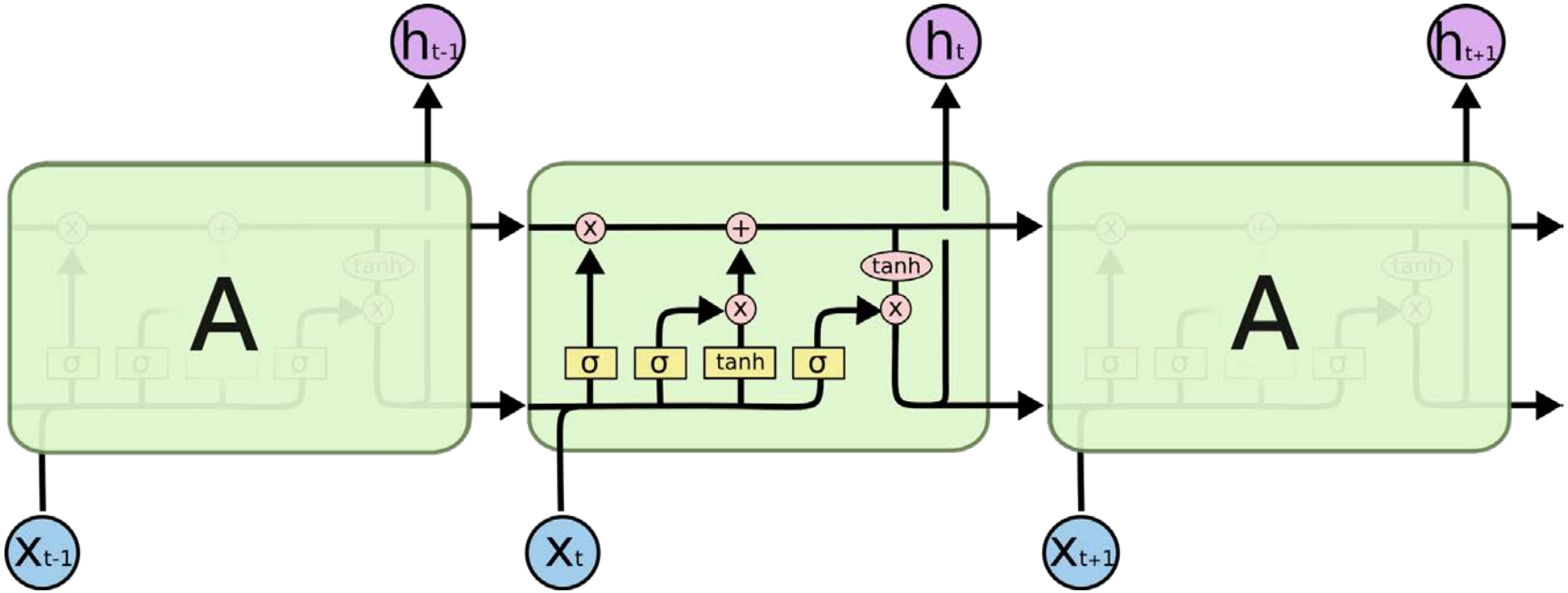


*< 1 *< 1 *< 1 ... *< 1

= 0.00000000000000000000001

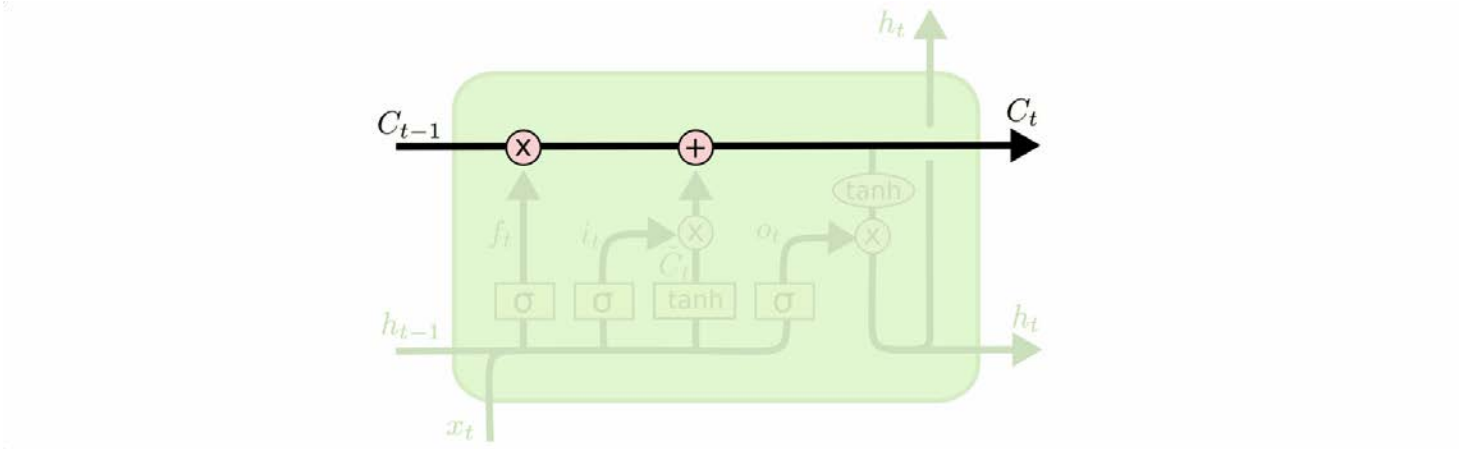
Recurrent Neural Networks

- Long Short-Term Memory networks (LSTMs)
  - Vanishing gradient problem 완화
  - 장기 의존성 학습문제 보완



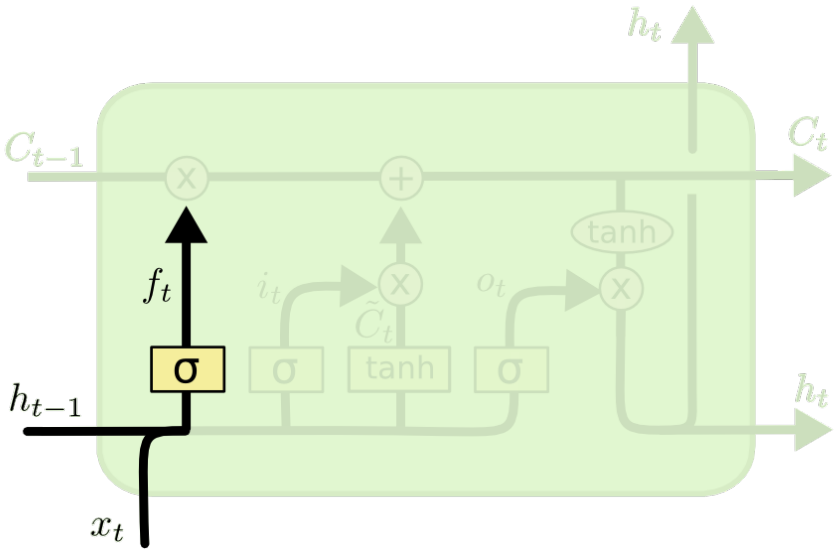
Recurrent Neural Networks

- LSTMs 핵심 아이디어
  - 셀 스테이트 (cell state) – 정보 전달 목적
    - 불필요한 정보 제거
    - 유용한 정보 추가



Recurrent Neural Networks

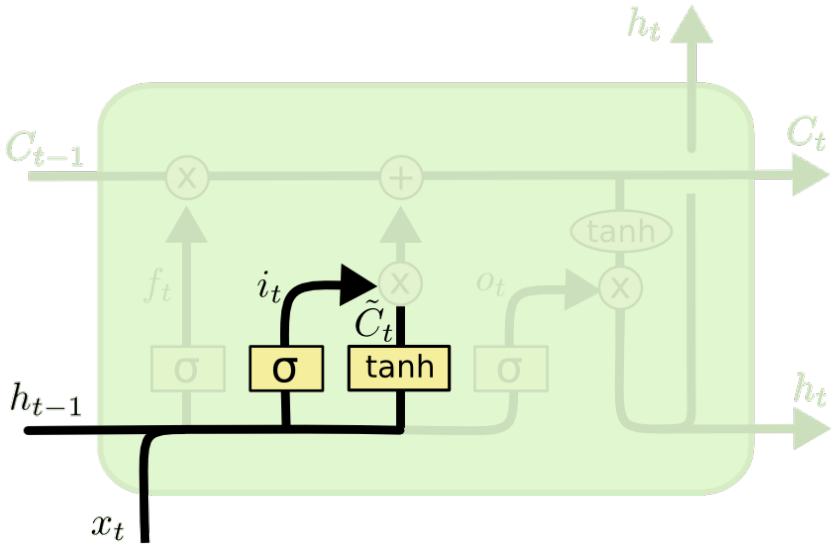
- LSTMs Step1
  - Forget gate layer
    - 어떤 정보를 셀 스테이트에서 제거할 것인지 결정



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

Recurrent Neural Networks

- LSTMs Step2
  - Input gate layer
    - 어떤 정보를 셀 스테이트에 **더해** 줄 것인지 결정

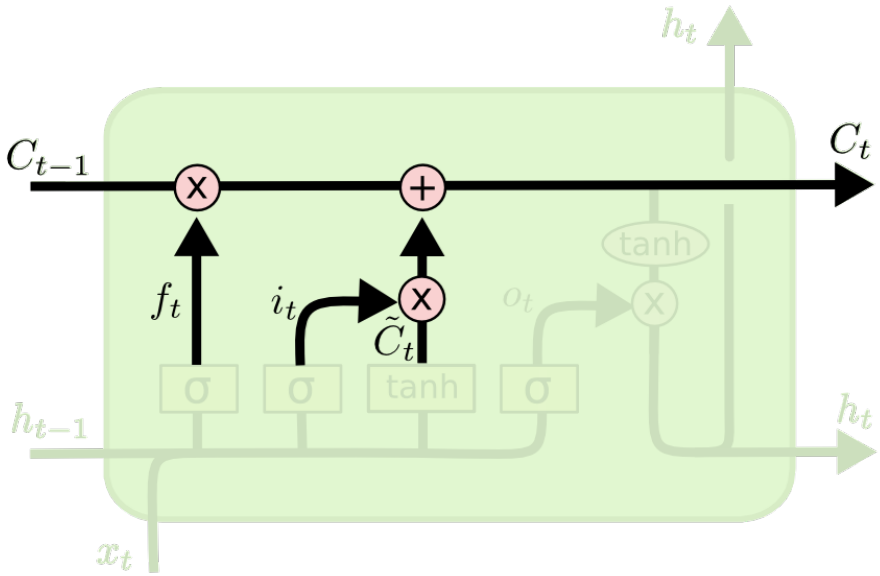


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

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

Recurrent Neural Networks

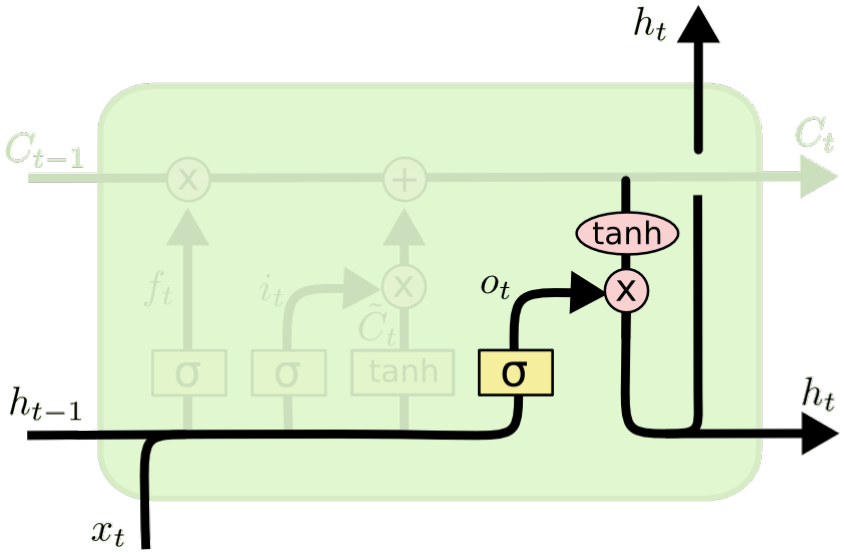
- LSTMs Step3
  - Update the cell state
    - 과거의 C_{t-1}을 새로운 C_t로 업데이트



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

Recurrent Neural Networks

- LSTMs Step4
  - Output gate layer
    - 셀 스테이트로부터 어떤 정보를 읽을 것인지 결정



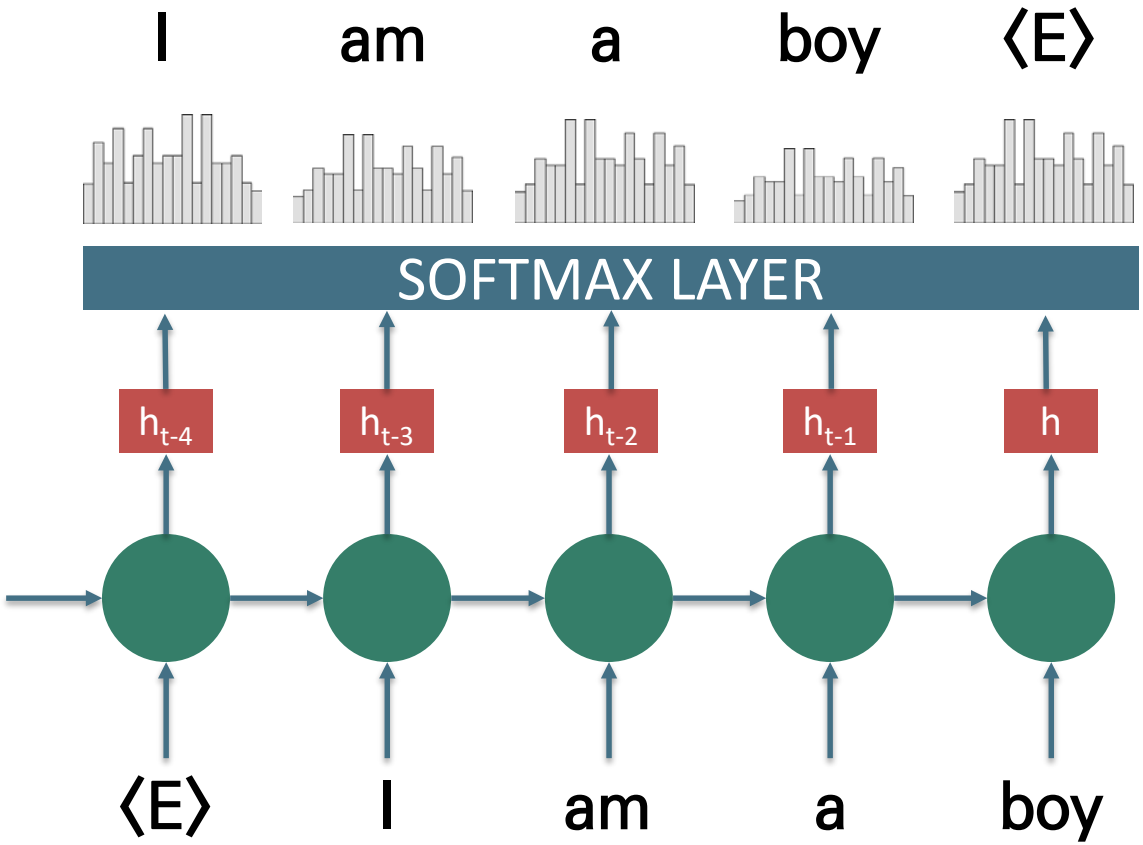
$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

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

Recurrent Neural Networks

- LSTMs의 다양한 변형
  - Peep hole
  - Forget + Input gate
  - Gated Recurrent Unit (GRU)

신경망 언어 모델



$$P(i, am, a, boy, \langle E \rangle) = P(i)*P(am|i)*P(a|i,am)*P(boy|i,am,a)*P(\langle E \rangle|i,am,a,boy)$$

LSTM 기반 언어모델 실습

LSTM based language model

- Training 과정
  - 학습데이터 (수만 문장 이상)
    - i am a boy .
    - sometimes to understand a word's…
    - at your dictionary we try to gib…
    - …
  - 단어 사전 구축
    - {i=1, am=2, a=3, boy=4, .=5, sometimes=6, …}
  - 문장 속 단어들 → 숫자들로 변환
    - 1 2 3 4 5
    - 6 7 8 9 3 10 …
    - …

LSTM based language model

- Training 과정
  - One-hot representation 변환
    - 1 2 3 4 5

One-hot vector representation

Training sentence

	⟨E⟩	i	am	a	boy	.	some	⋯	⟨pad⟩
⟨E⟩	1	0	0	0	0	0	0	...	0
i	0	1	0	0	0	0	0	...	0
am	0	0	1	0	0	0	0	...	0
a	0	0	0	1	0	0	0	...	0
boy	0	0	0	0	1	0	0	...	0
.	0	0	0	0	0	1	0	...	0
⟨E⟩	0	0	0	0	0	0	0	...	1
⟨pad⟩	0	0	0	0	0	0	0	...	1
⋮	⋮								
⟨pad⟩	0	0	0	0	0	0	0	...	1

LSTM based language model

- Training 과정
  - Word-embedding 변환
    - 1 2 3 4 5

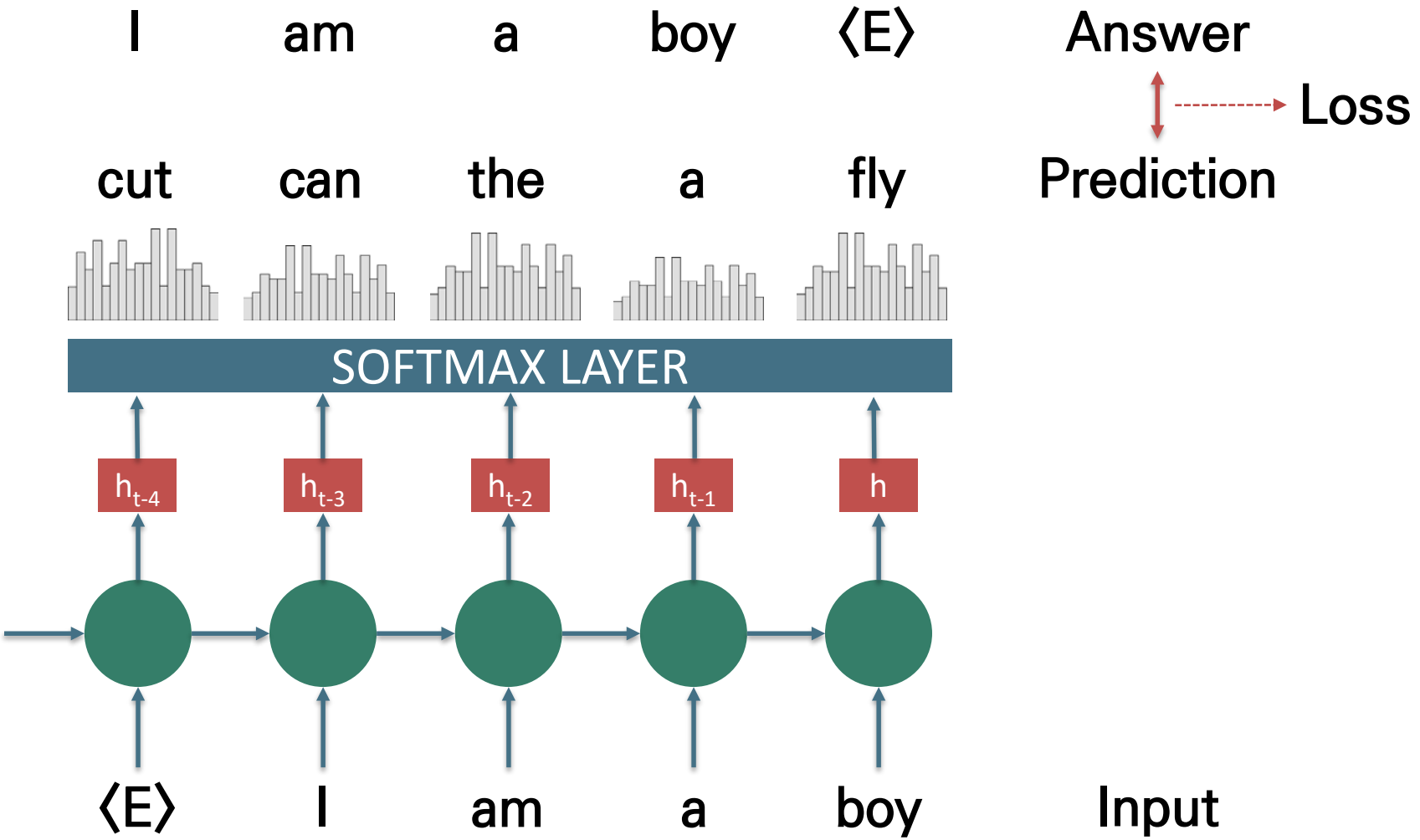
Word-embedding

Training sentence

	?	?	?	?	?	?	?	?	?
⟨E⟩	0.24	0.15	0.58	0.94	0.14	0.25	0.33	0.85	0.15
i	0.11	0.78	0.91	0.17	0.64	0.75	0.64	0.87	0.36
am	0.78	0.91	0.33	0.87	0.36	0.87	0.36	0.25	0.33
a	0.85	0.15	0.36	0.64	0.78	0.64	0.75	0.87	0.36
boy	0.25	0.33	0.33	0.85	0.64	0.75	0.33	0.64	0.75
.	0.91	0.33	0.64	0.58	0.94	0.25	0.33	0.15	0.58
⟨E⟩	0.64	0.33	0.85	0.64	0.33	0.75	0.64	0.91	0.17
⟨pad⟩	0	0	0	0	0	0	0	0	0
⋮	⋮								
⟨pad⟩	0	0	0	0	0	0	0	0	0

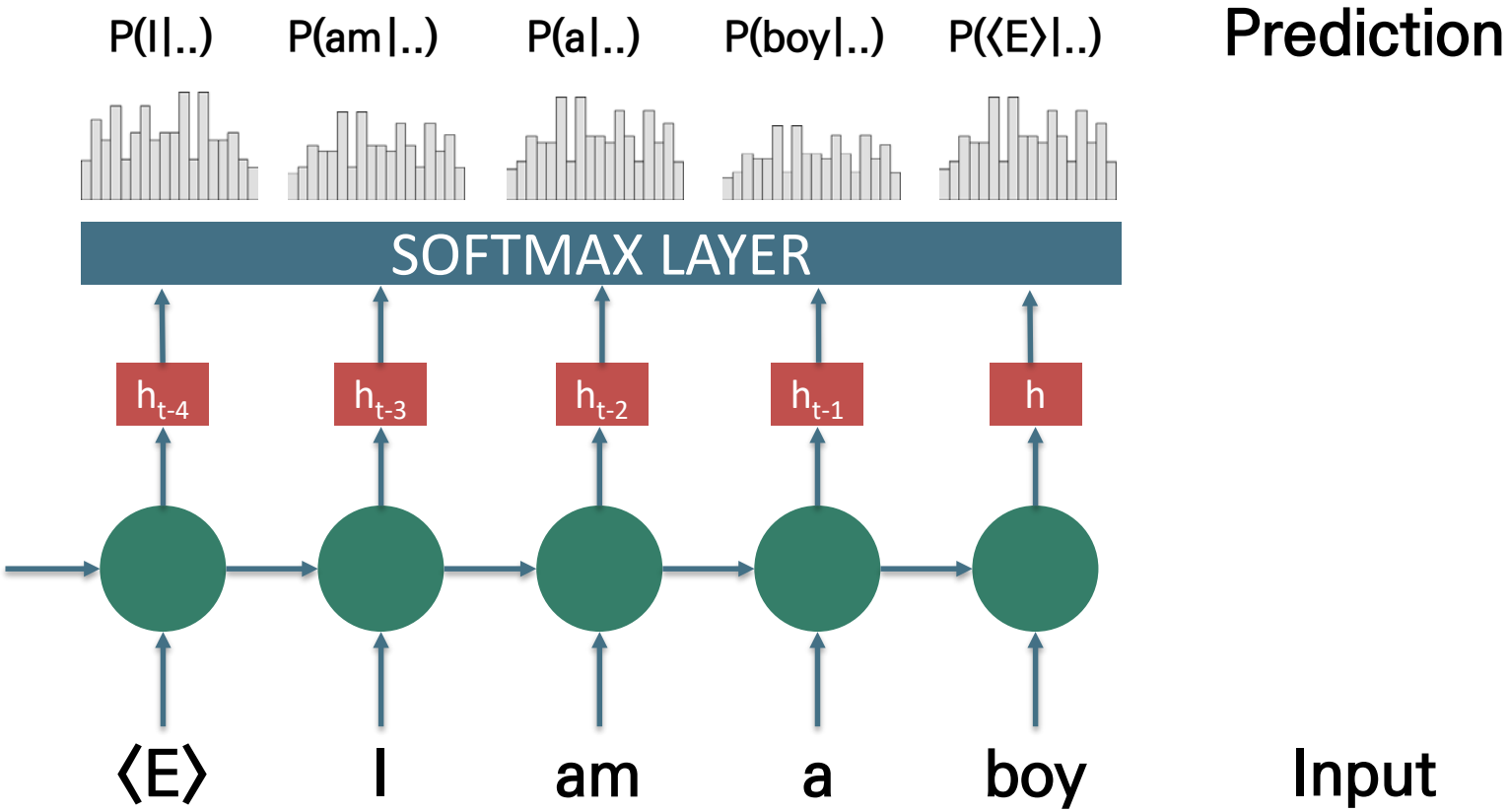
LSTM based language model

- Training 과정



LSTM based language model

- Testing 과정



- P(⟨E⟩, i, am, a, boy) =
  P(i) * P(am|i) * P(a|i,am) * P(boy|i,am,a) * P(⟨E⟩|i,am,a,boy)

CODE REVIEW

언어모델 실습

- 환경 구축
  - Pytorch 1.4.0
  - Pip3 install torch

언어모델 실습

- Data.py
  - 학습 데이터를 받아 Dictionary와 Corpus 구성
  - Corpus.dictionary.word2idx[]로 사전에 있는 단어의 인덱스 획득
  - Corpus.dictionary.idx2word[]로 인덱스에서 단어로 변환

- Model.py
  - LSTM 기반의 LM 모델
  - 실제 모델의 작동 방식 구현
  - Forward() : input과 hidde을 받아 outpu과 hidden 생성

언어모델 실습

- Main.py
  - 모델의 학습 및 저장
  - 학습 데이터를 Corpus로 변환하고 해상 Corpus를 mini-batch로 분할
  - 모델을 Loss를 계산하여 Backprop

- Seq_prob.ipynb
  - 실습 함수 : 빈 칸 채워 실제 확률 계산
  - 모델을 이용하여 문장의 확률 계산

언어모델 실습

* 문장 확률 계산하기

- P(⟨E⟩, i, am, a, boy) =
  P(i)*P(am|i)*P(a|i,am)*P(boy|i,am,a)*P(⟨E⟩|i,am,a,boy)
- Def seq_prob(seq) 함수 완성
  1. Seq 속 단어들 index로 바꾸기
  2. 모델에 들어갈 hidden 값 초기화
  3. 모델로부터 input과 hidden에 대한 결과값 얻기
  4. 각 단어별 확률 계산
  5. 최종 seq 확률(아래) 계산 하여 return 하기
     P(i)*P(am|i)*P(a|I,am)*P(boy|I,am,a)*P(⟨E⟩|I,am,a,boy)
  6. 결과 출력 확인 (아래 두 문장의 확률 비교)
     print(' P(I am a boy) =', seq_prob(['i', 'am', 'a', 'boy']))
     print('P(i boy am a) =', seq_prob(['i', 'boy', 'am', 'a']))

Q & A