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(<u>high</u> winds tonight) > P(<u>large</u> winds tonight)
  - 철자 교정
    - P(about fifteen <u>minutes</u> from) > P(about fifteen <u>minuets</u> from)
  - 기타 등등…

언어 모델

- 접근 방법
  - $P(\text{Today is Wednesday})$

    $= P(\text{Today})P(\text{is}|\text{Today})P(\text{Wednesday}|\text{is, Today})$

    ↓ (a.k.a Auto-regressive)

  - $P(\text{W}) = P(w_1, w_2, w_3, \dots, w_n)$

    $= P(w_1)P(w_2|w_1)P(w_3|w_2, w_1) \dots P(w_n|w_{n-1}, w_{n-2}, \dots, w_1)$

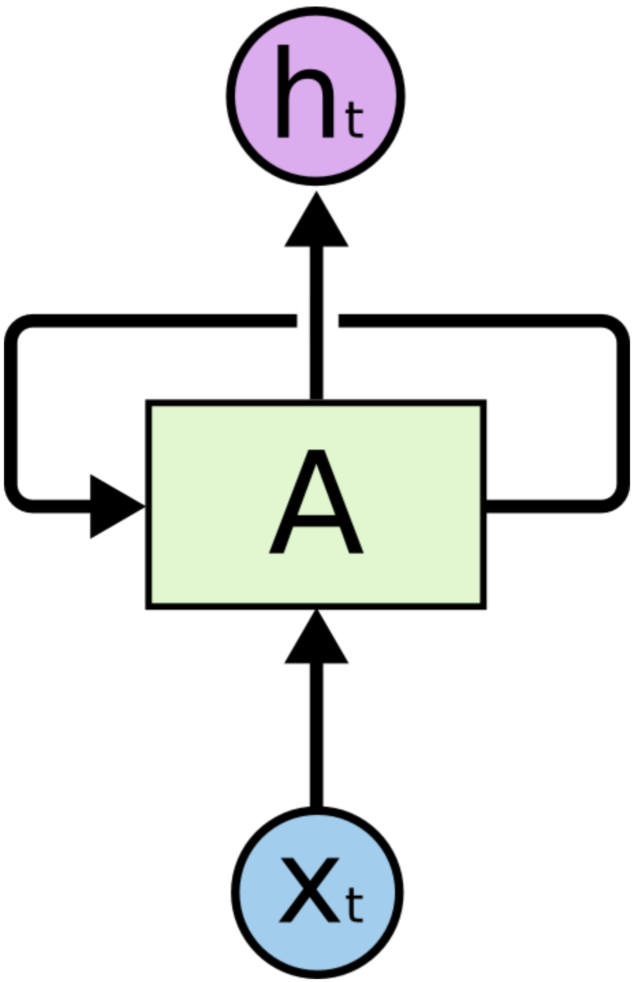
    $$= \prod_{i=1}^{n} P(w_i|w_1^{i-1})$$

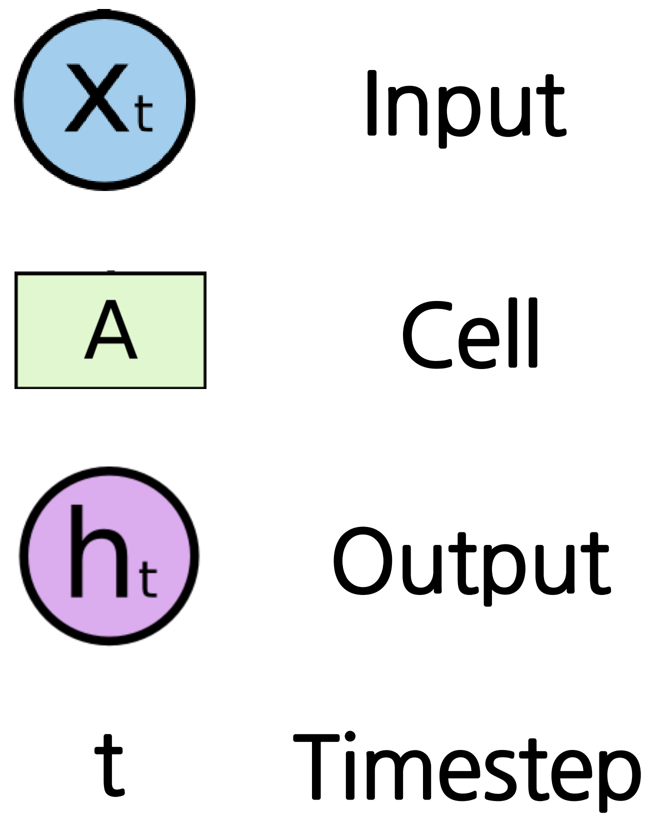
언어 모델

- 통계기반 언어 모델
  - N-gram 언어 모델
- 신경망 기반 언어 모델 - Vector space model
  - Recurrent neural network 기반 언어 모델

Recurrent Neural Networks

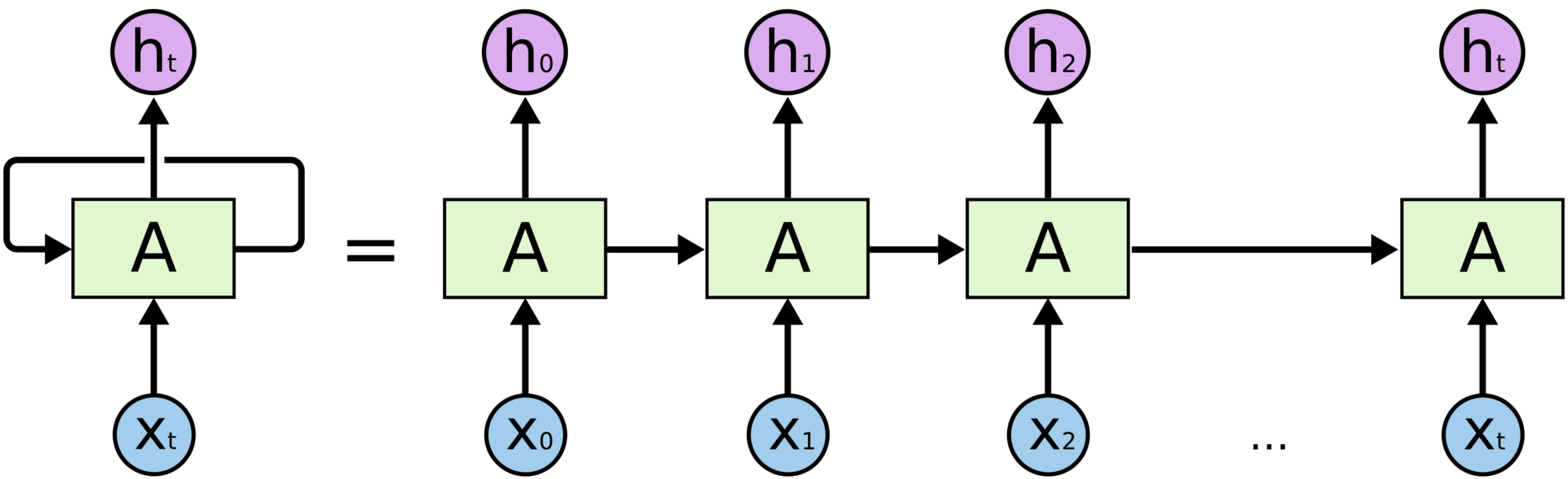
- 순환신경망 (Recurrent Neural Networks; RNNs)





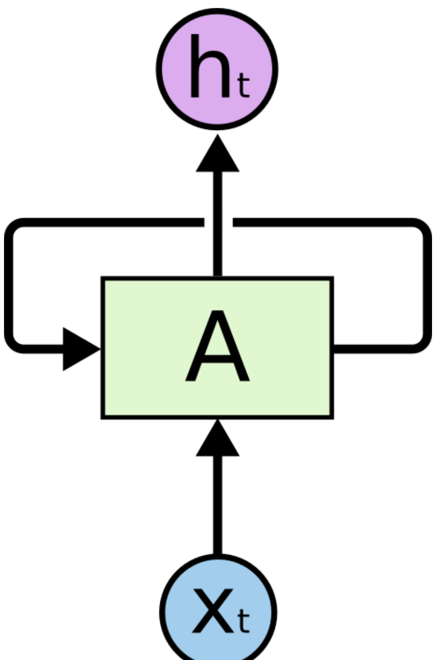
Recurrent Neural Networks

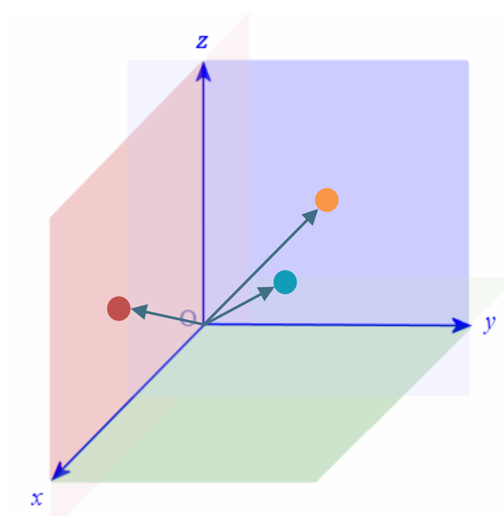
- 순환신경망 (Recurrent Neural Networks; RNNs)



Recurrent Neural Networks

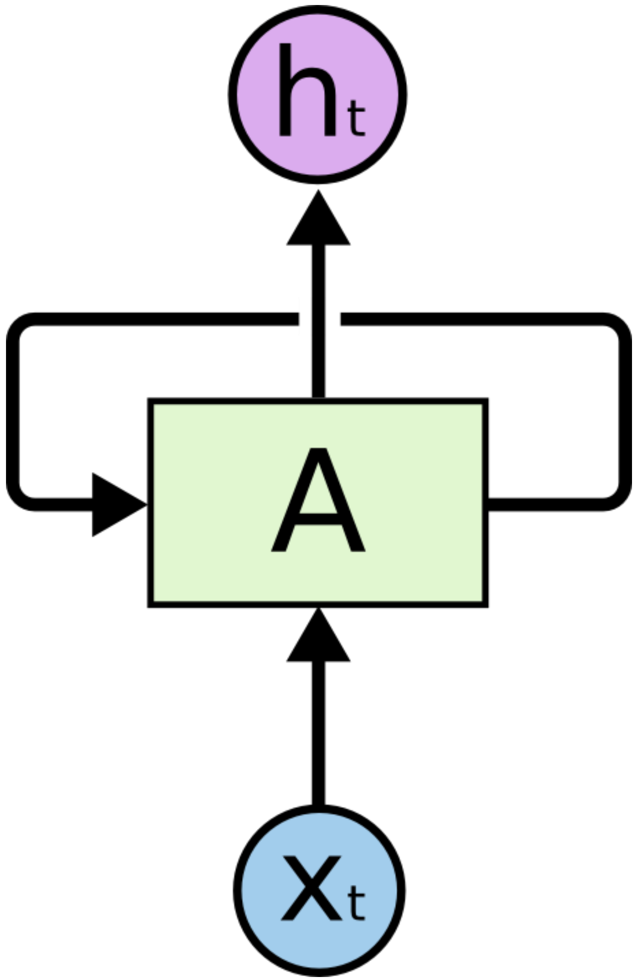
- 순환신경망 (Recurrent Neural Networks; RNNs)
  - 무작위 길이의 열 → 고정된 길이의 벡터 표현
    - I am a boy
    - Sometimes to understand a word's…
    - At your dictionary we try to gib…

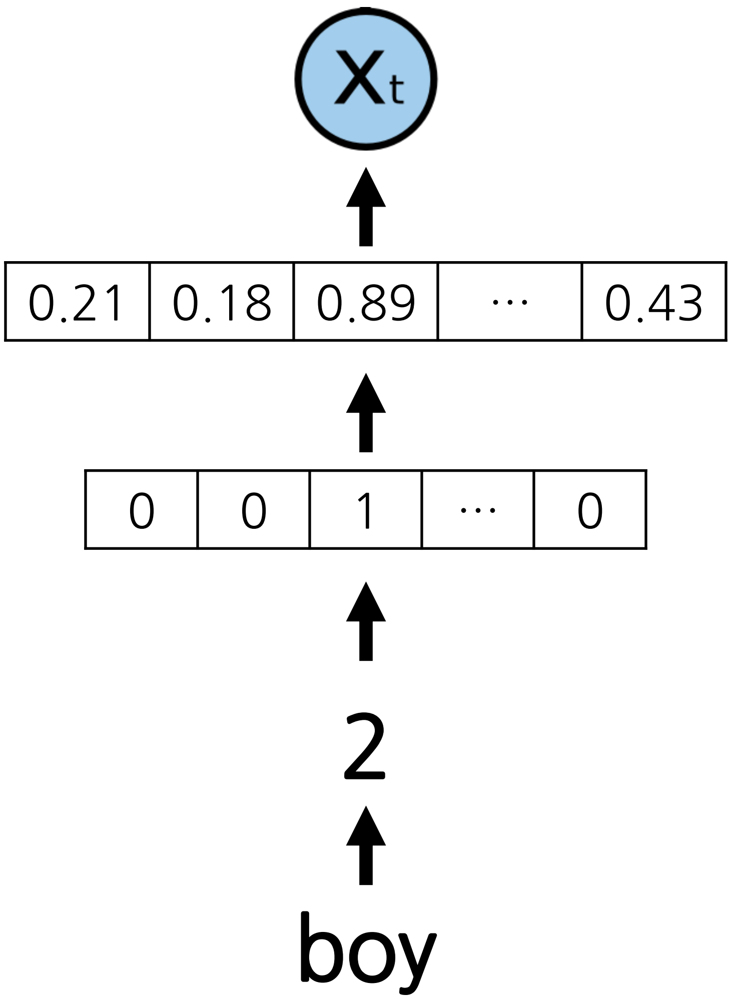




Recurrent Neural Networks

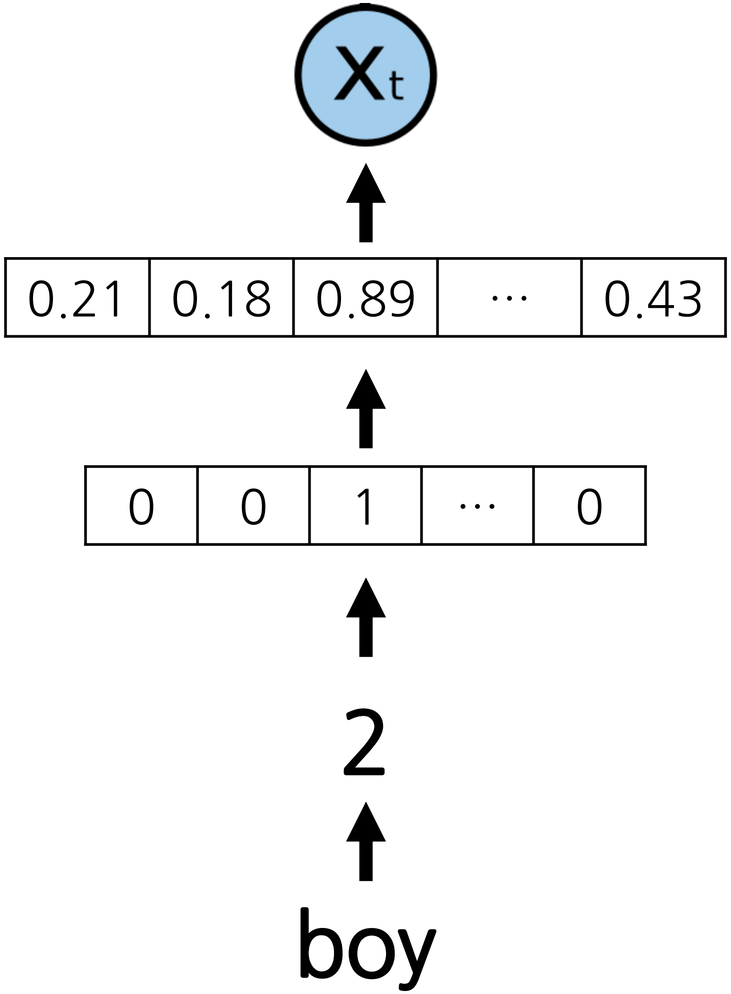
- RNN의 입력

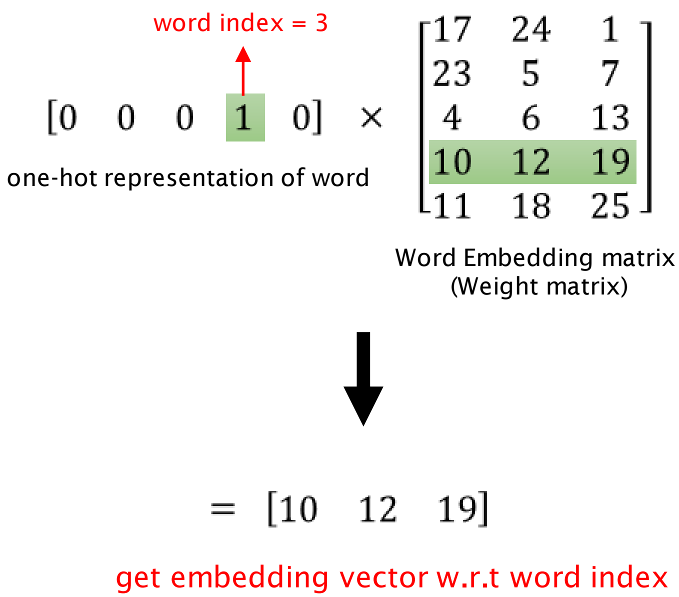




Recurrent Neural Networks

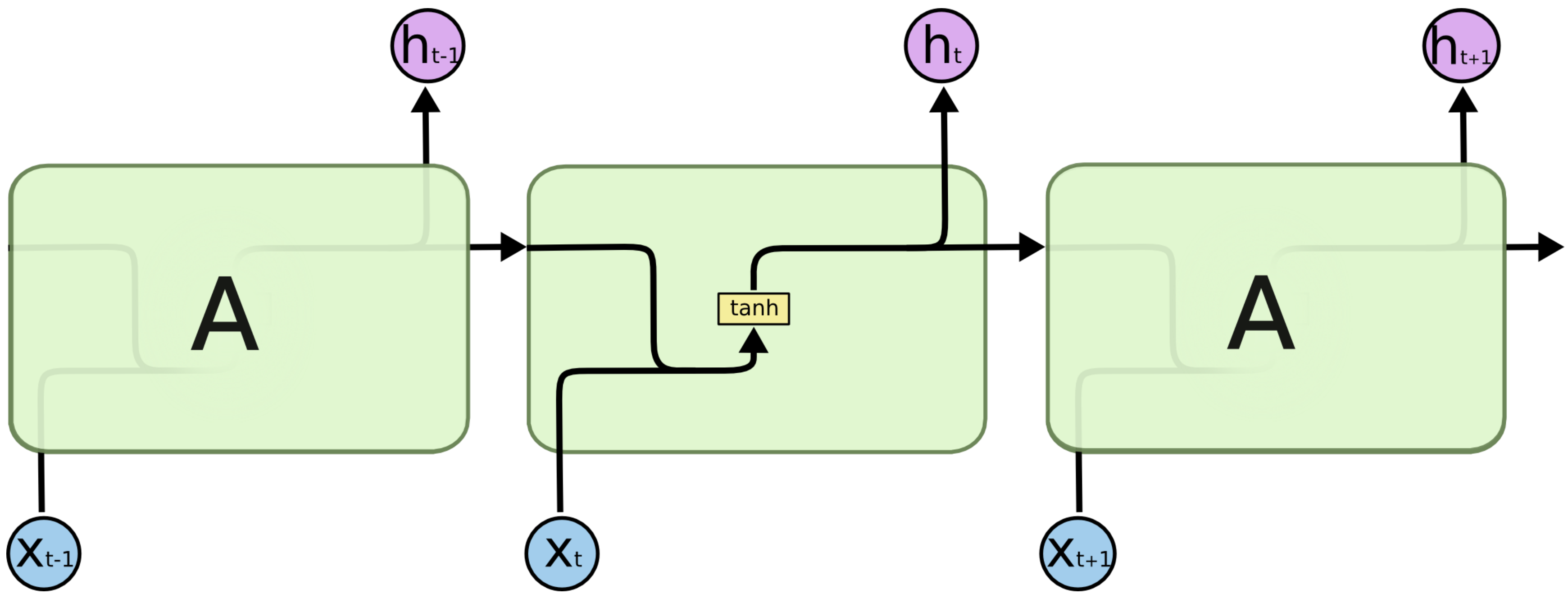
- RNN의 입력: Embedding Layer (Word to Vector)





Recurrent Neural Networks

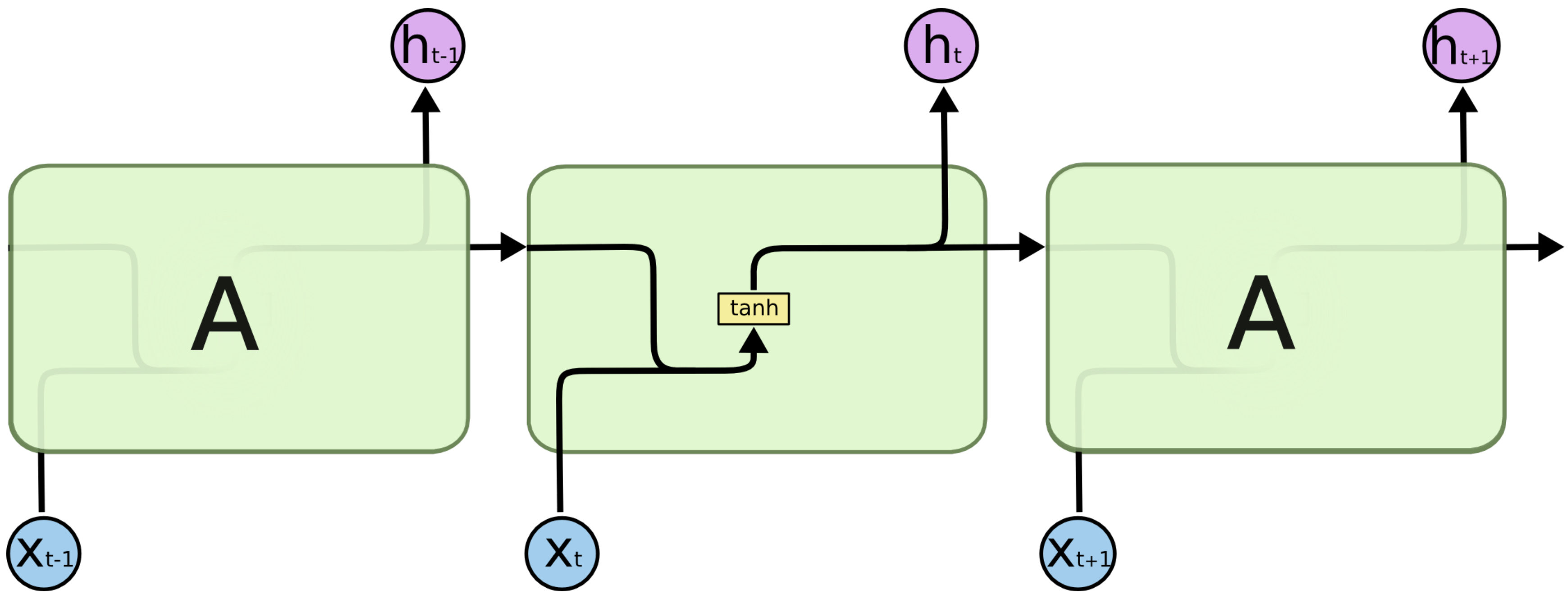
- RNN 출력



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

Recurrent Neural Networks

- Timestep마다 다른 Weight? Or weight sharing?



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

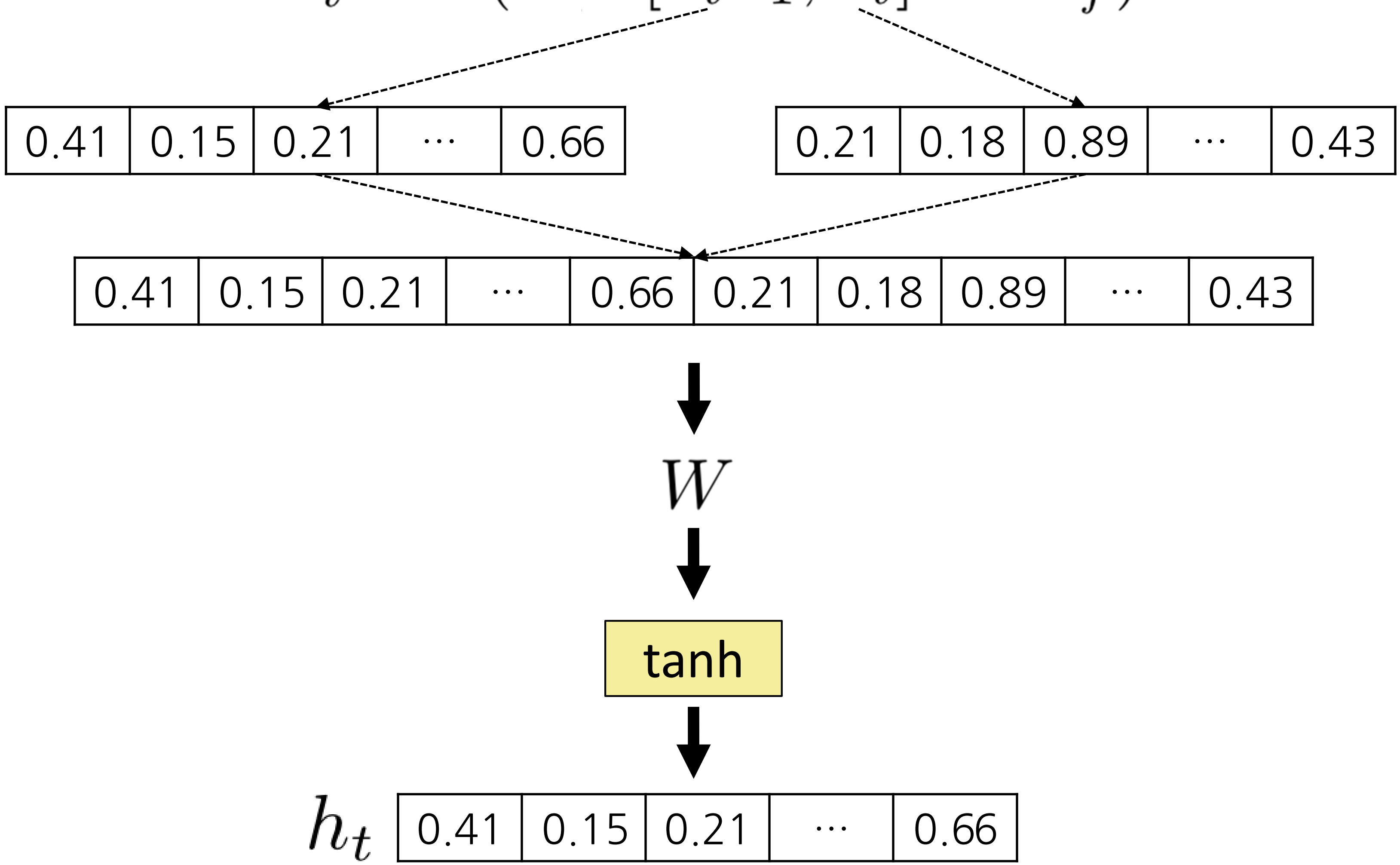
Recurrent Neural Networks

- Timestep마다 동일한 weight 공유
  - 학습 파라미터의 수 감소
  - 네트워크가 학습하지 못한 입력열에 대한 일반화 용이 (Overfitting 감소)
  - 가변길이 입력열에 대한 모델링 가능
  - on monday it was snowing ≈ it was snowing on Monday

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

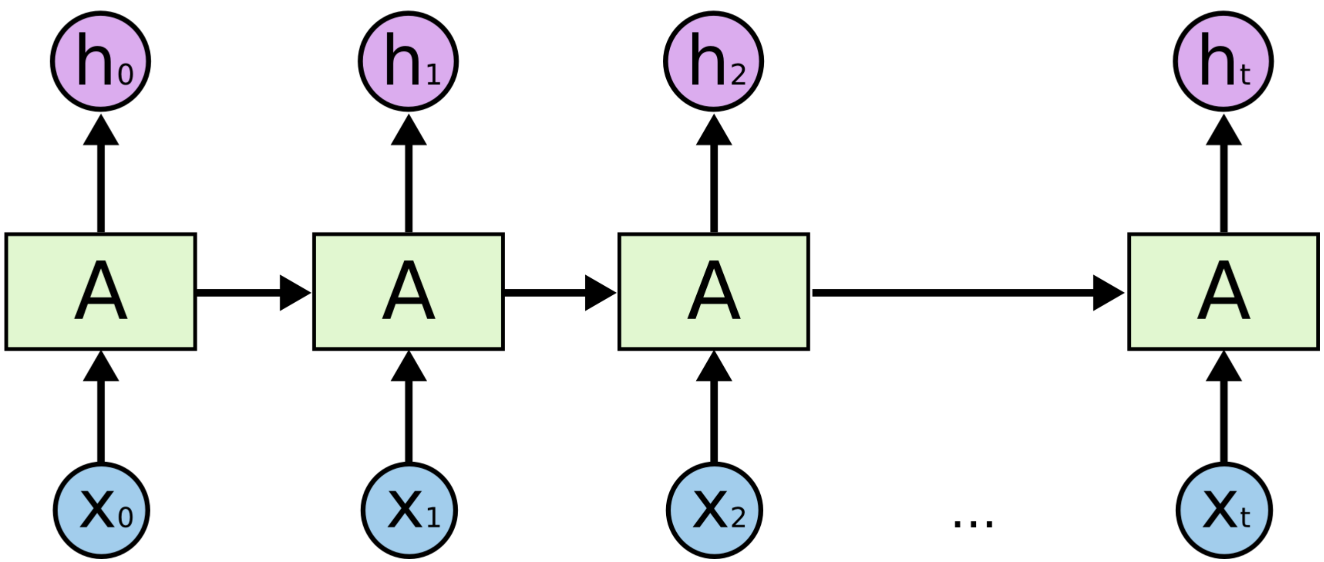
Recurrent Neural Networks

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



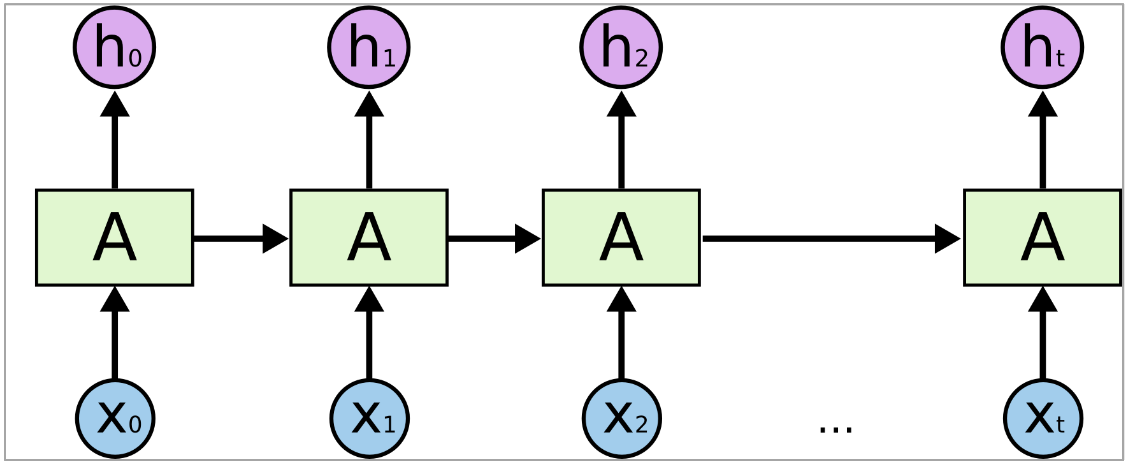
Recurrent Neural Networks

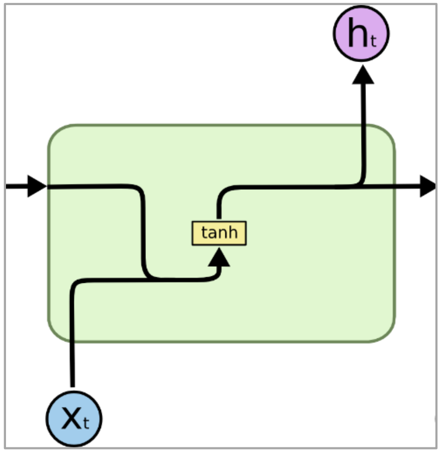
- 불행하게도, 길이가 긴 열 학습 어려움
  - Vanishing gradient problem



Recurrent Neural Networks

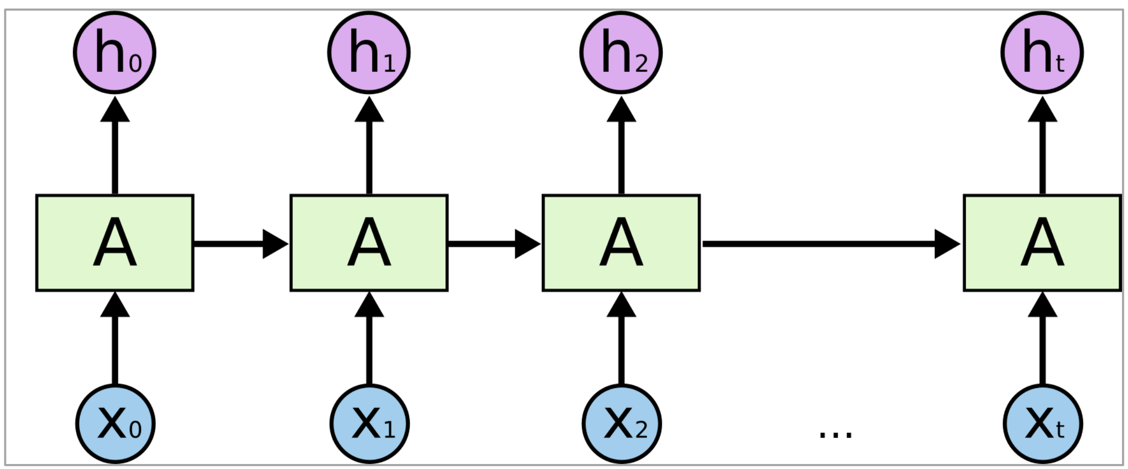
- 불행하게도, 길이가 긴 열 학습 어려움
  - Vanishing gradient problem

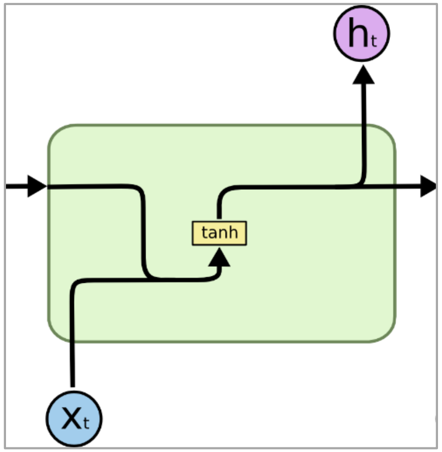




Recurrent Neural Networks

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

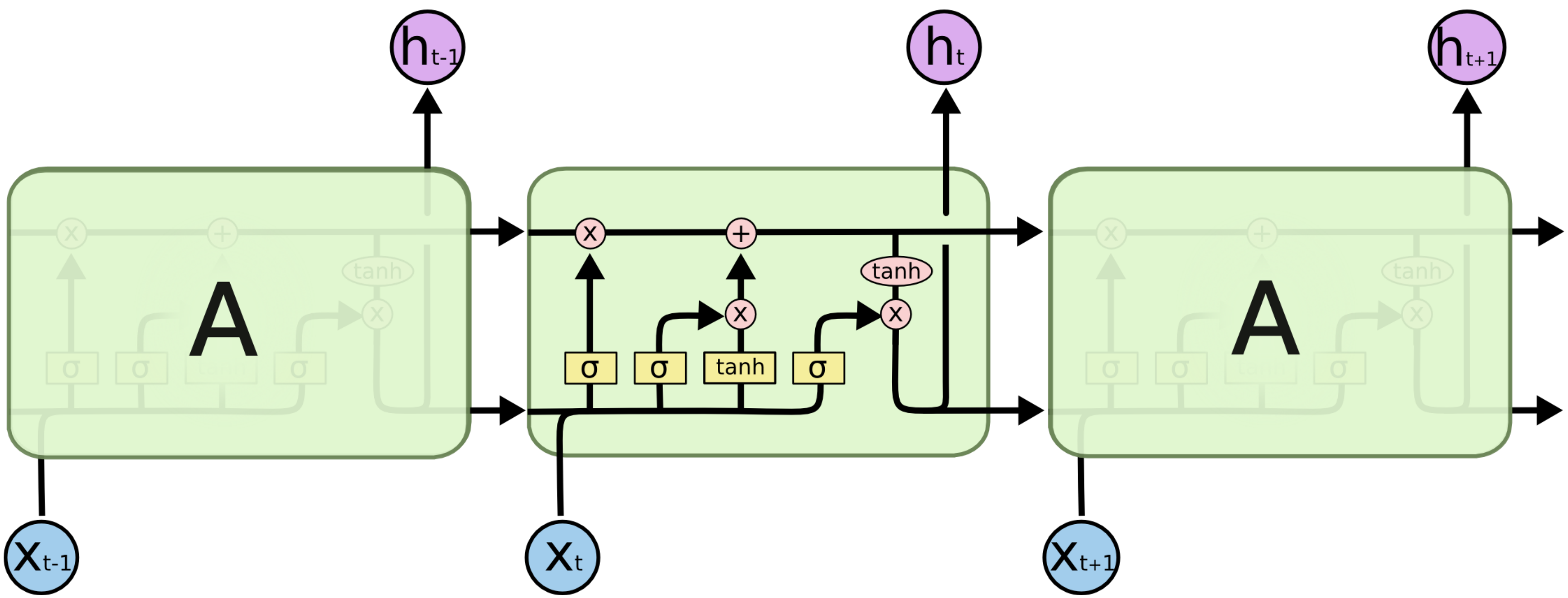




$$* < 1 \quad * < 1 \quad * < 1 \quad \cdots \quad * < 1$$
$$= 0.0000000000000000000001$$

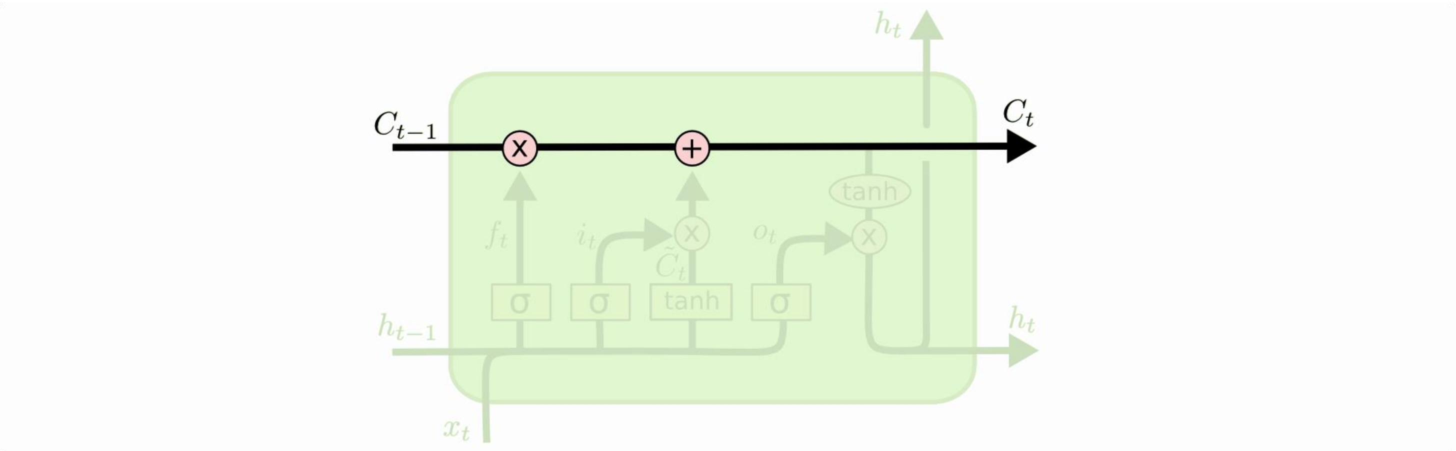
Recurrent Neural Networks

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



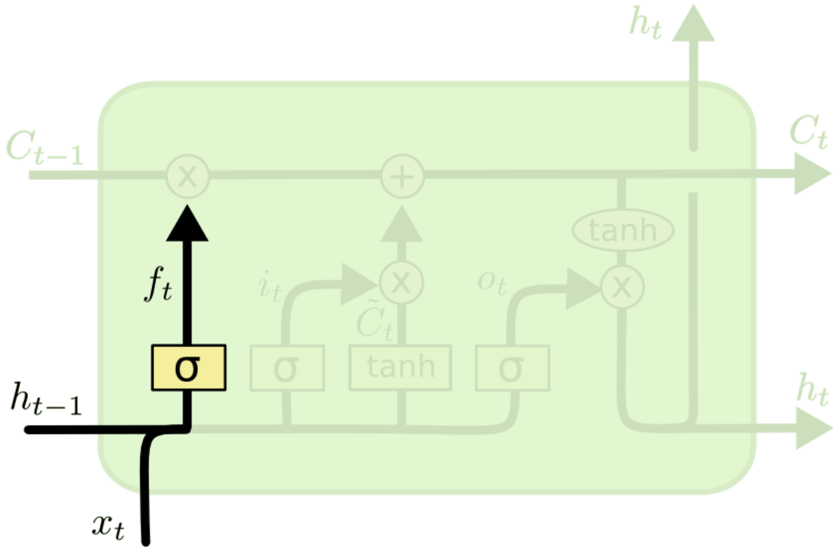
Recurrent Neural Networks

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



Recurrent Neural Networks

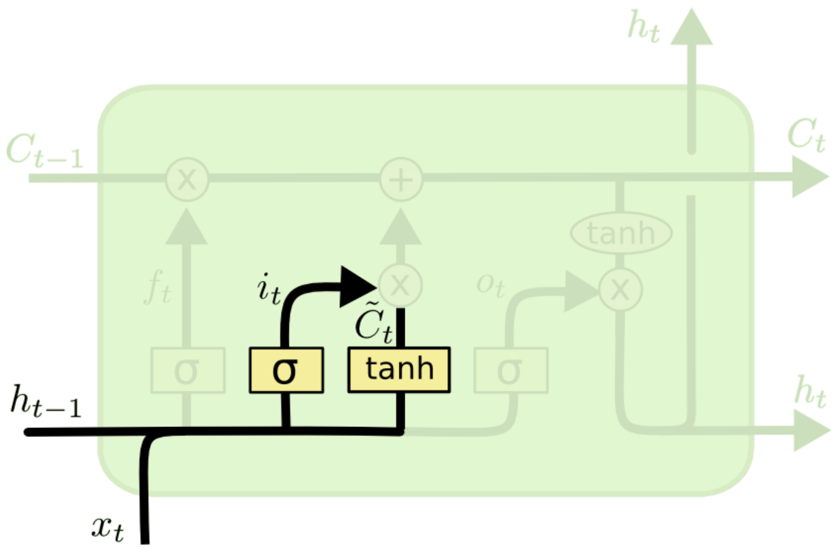
- LSTMs Step1
  - Forget gate layer
    - 어떤 정보를 셀 스테이트에서 제거할 것인지 결정



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

Recurrent Neural Networks

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

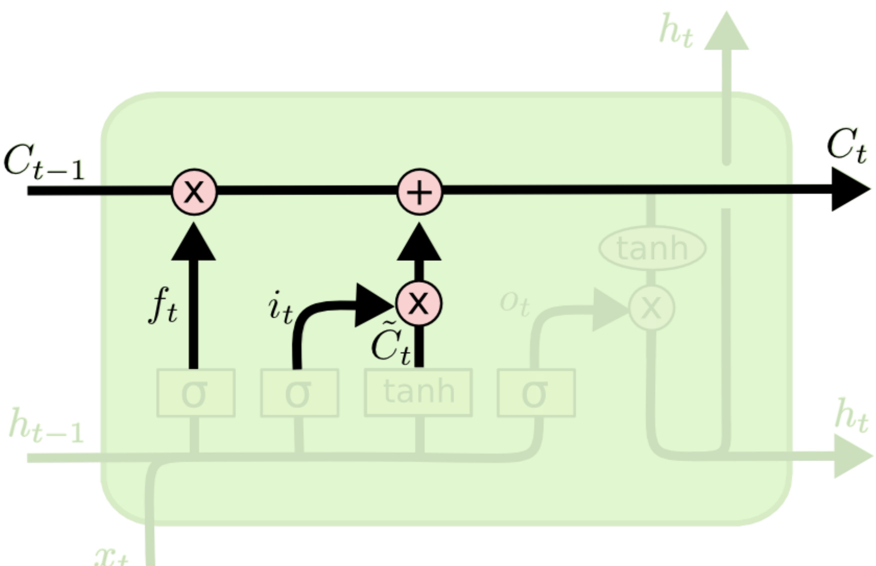


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$

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

Recurrent Neural Networks

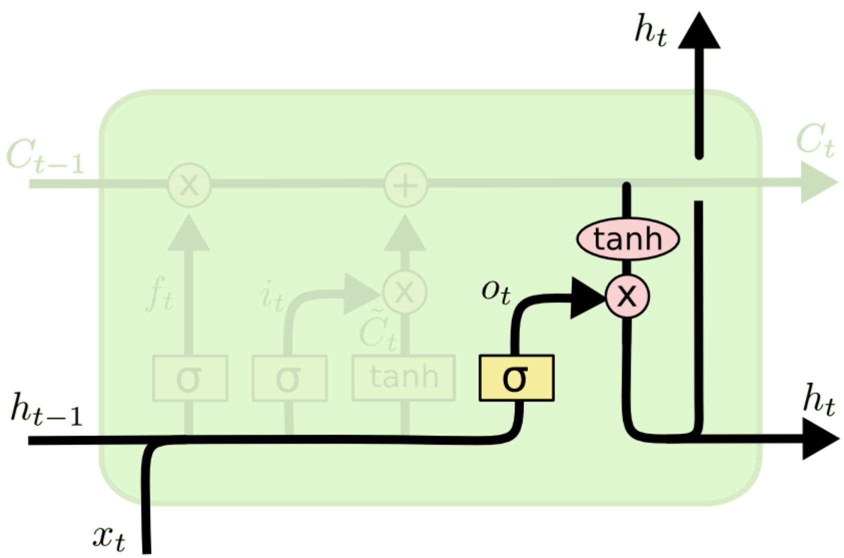
- LSTMs Step3
  - Update the cell state
    - 과거의 C_{t-1}을 새로운 C_t로 업데이트



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

Recurrent Neural Networks

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



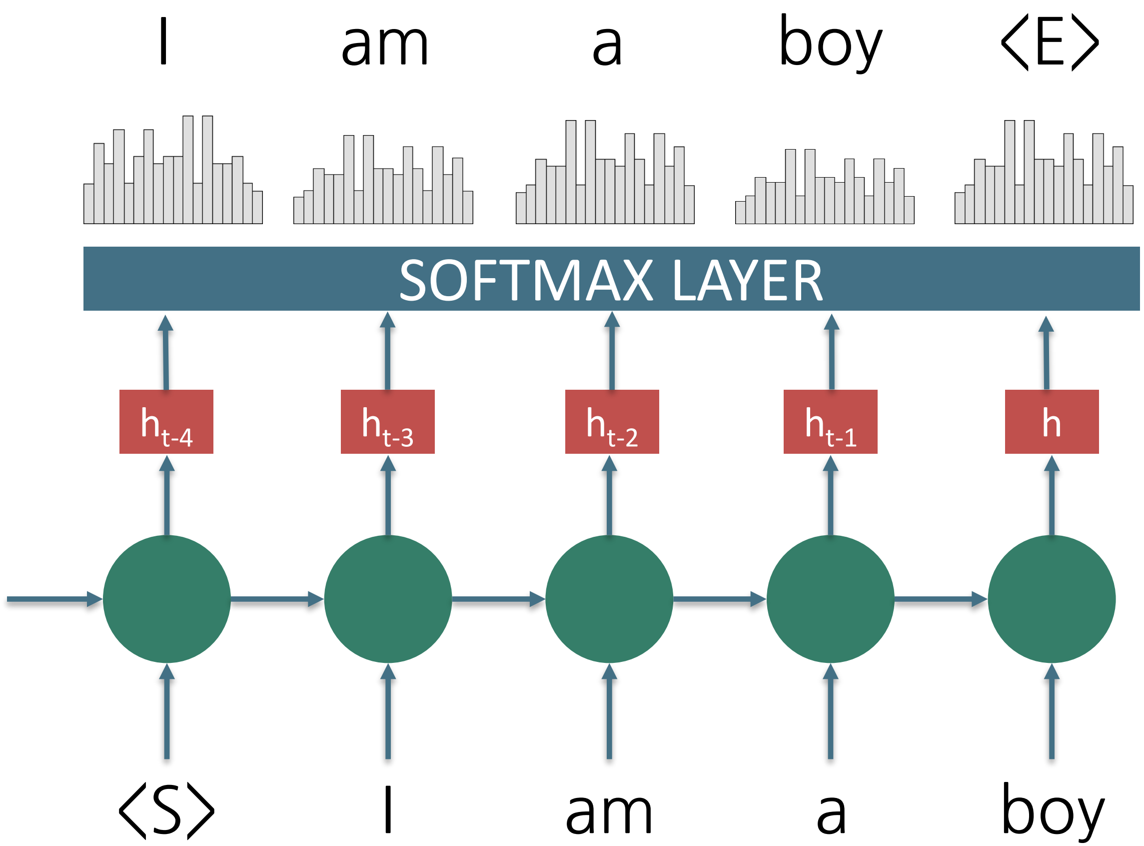
$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$

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

Recurrent Neural Networks

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

신경망 언어 모델



$$P(\langle S\rangle \text{ i, am, a, boy}) = P(i|\langle S\rangle) * P(am|I,\langle S\rangle) * P(a|i,am,\langle S\rangle)$$
$$*P(boy|i,am,a,\langle S\rangle) * P(\langle E\rangle|i,am,a,boy,\langle S\rangle)$$

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

- Batching
- input

<S>	I	am	a	boy	.	<pad>	<pad>
<S>	sometimes	to	understand	a	word	.	<pad>
<S>	we	try	to	build	a	dictionary	.

- Output (Target)

I	am	a	boy	.	<E>	<pad>	<pad>
sometimes	to	understand	a	word	.	<E>	<pad>
we	try	to	build	a	dictionary	.	<E>

LSTM based language model

- Batching
- input

7	1	2	3	4	5	0	0
7	6	to	7	a	8	5	0
7	9	10	11	12	3	13	5

{<pad>=0, i=1, am=2, a=3, boy=4, .=5, sometimes=6, <S>=7, <E>=8, …}

- Output (Target)

1	2	3	4	5	8	0	0
6	to	7	a	8	5	8	0
9	10	11	12	3	13	5	8

LSTM based language model

- Training 과정
  - One-hot representation 변환
    - word idx: 1 2 3 4 5 6

One-hot vector representation

Training sentence

<S>	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
<pad>	0	0	0	0	0	0	1	...	0
<pad>	0	0	0	0	0	0	1	...	0
⋮	⋮								
<pad>	0	0	0	0	0	0	1	...	0

LSTM based language model

- Training 과정
  - Word-embedding 변환
    - word idx: 1 2 3 4 5 6

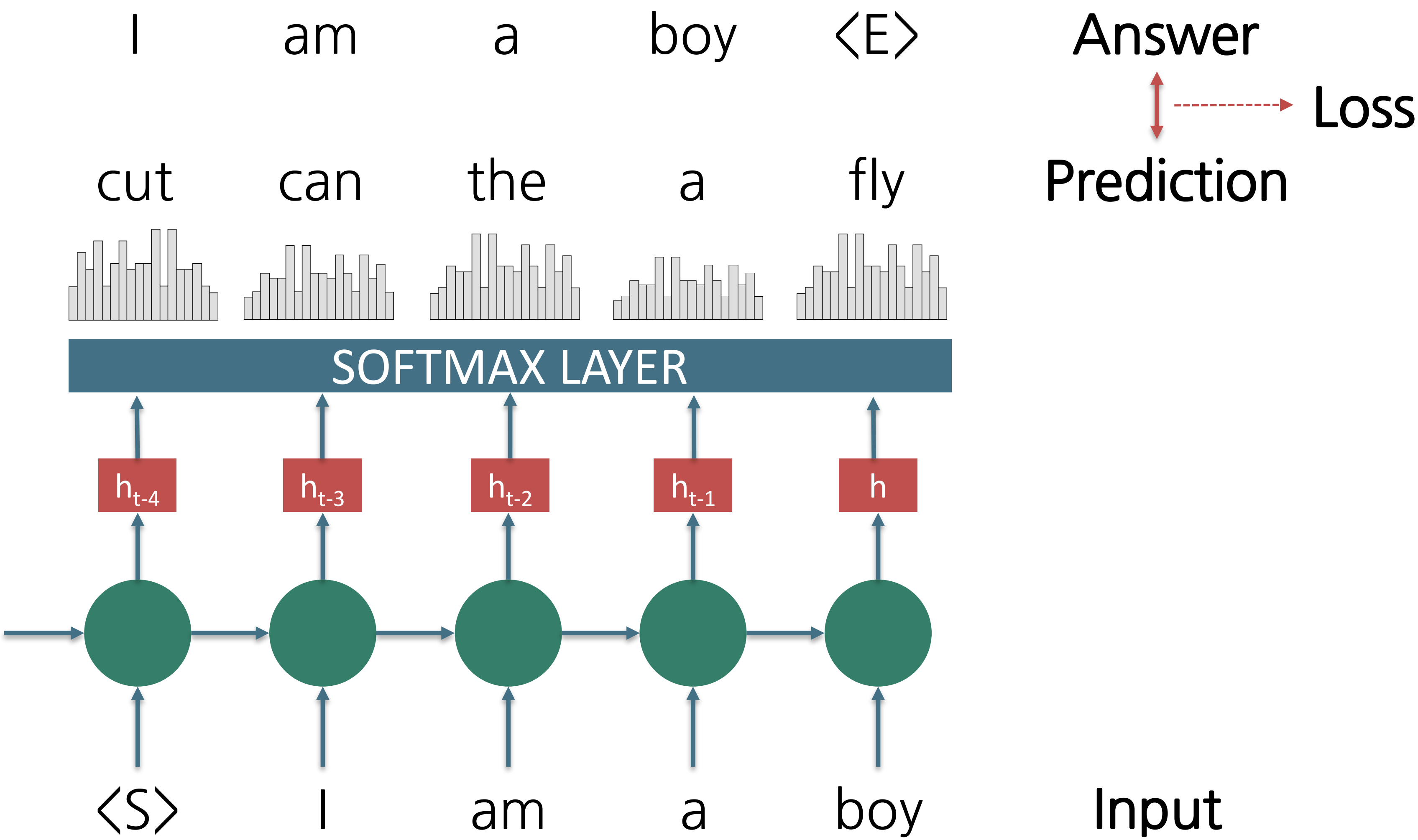
Word-embedding

Training sentence

<S>	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
<pad>	0	0	0	0	0	0	0	0	0
<pad>	0	0	0	0	0	0	0	0	0
⋮	⋮								
<pad>	0	0	0	0	0	0	0	0	0

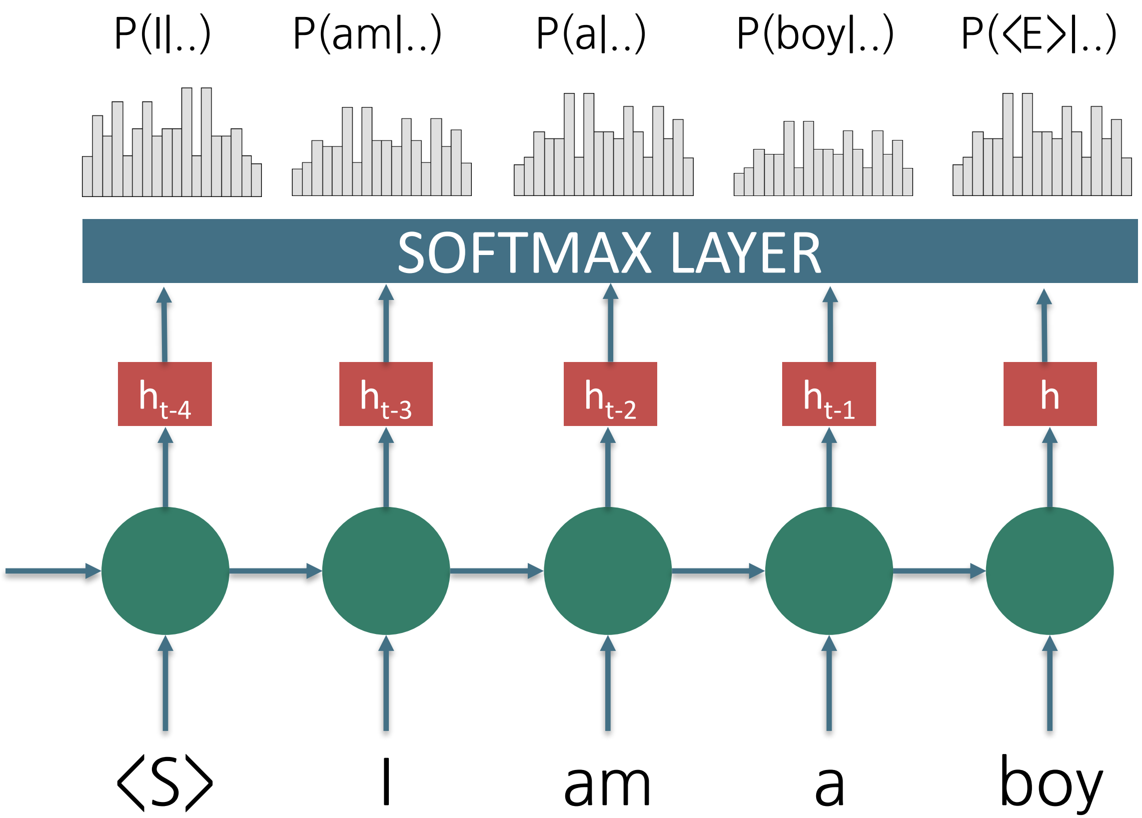
LSTM based language model

- Training 과정



LSTM based language model

- Testing 과정



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

CODE REVIEW

언어모델 실습

- NLTK 설치
  - sudo pip3 install nltk

언어모델 실습

1. 언어 모델 학습하기 isTrain=True
2. 언어 모델 학습 완료 후 isTrain=False
3. 주어진 문장 확률 분포 계산
   - P(〈S〉, i, am, a, boy)
     = P(i|〈S〉) ∗ P(am|I,〈S〉) ∗ P(a|i,am,〈S〉) ∗ P(boy|i,am,a,〈S〉)
     ∗ P(〈E〉|i,am,a,boy,〈S〉)
   - def pred_sent_prob(sent) 함수 완성
     1. sent 의 단어 index 변환
     2. LSTM 에 입력 전달 후, output 출력
     3. output 값에서 위의 각 단어의 확률 값 (Log 확률) 추출
        : (P(i), P(am), P(a), P(boy), P(〈E〉)
     4. 각 단어의 Log 확률의 **합** return (Log 확률 합 ≃ 확률 곱)
     5. 결과 출력 확인 (아래 두 문장의 확률 비교)
        print('P(i am a boy) =', pred_sent_prob([[‘i', ‘am', ‘a', ‘boy']])
        print('P(i boy am a) =', pred_sent_prob([[‘i', ‘boy', ‘am', ‘a']])

Q & A