KOREA UNIVERSITY

DSBA CS224n 2021 Study

[Lecture 05] Language Models and RNNs
[Lecture 06] Simple and LSTM RNNs

DSBA
Data Science & Business Analytics

고려대학교 산업경영공학과

Data Science & Business Analytics Lab

발표자 : 고유경

00 발표 목차

Contents

1. Language Model
2. RNN Language Model
3. Problem of RNN
4. LSTM
5. Conclusion

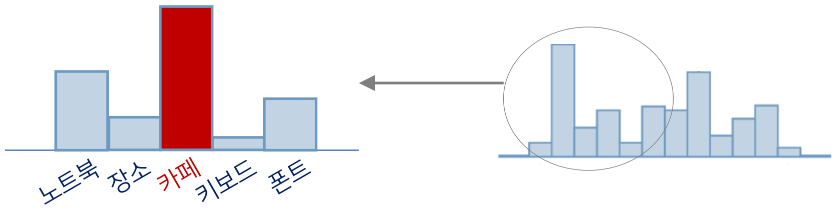
01 Language Model

Description

- 확률 분포를 기반으로 주어진 문맥(sequence) 이후에 위치할 단어 예측
  - 시퀀스의 결합 확률 by multiplication rule

$$P(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(T)}) = P(\boldsymbol{x}^{(1)}) \times P(\boldsymbol{x}^{(2)} | \boldsymbol{x}^{(1)}) \times \cdots \times P(\boldsymbol{x}^{(T)} | \boldsymbol{x}^{(T-1)}, \ldots, \boldsymbol{x}^{(1)})$$
$$= \prod_{t=1}^{T} P(\boldsymbol{x}^{(t)} | \boldsymbol{x}^{(t-1)}, \ldots, \boldsymbol{x}^{(1)})$$

  - $P\left(x^{(3)} \middle| \text{코딩하기}, \text{좋은}\right)$



01 Language Model

N-gram LM, Neural LM

N-gram Language Model

$\boldsymbol{P}$(코딩하기, 좋은, 카페, 추천)

uni-gram

P(코딩하기) P(좋은) P(카페) P(추천)

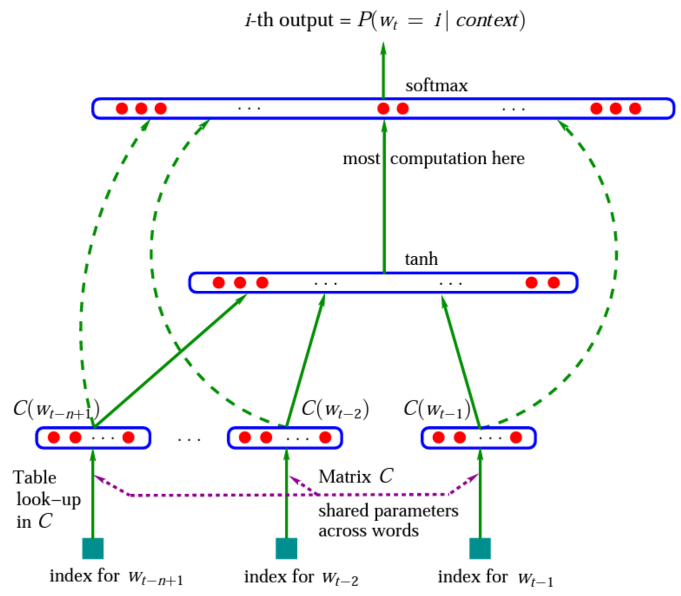
bi-gram

P(코딩하기) P(좋은| 코딩하기) P(카페| 좋은) P(추천| 카페)

$$P(\text{카페}|\ \text{코딩하기, 좋은}) = \frac{count(\text{코딩하기, 좋은, 카페})}{count(\text{코딩하기, 좋은})}$$

- ✓ 이전에 등장한 n개의 단어를 바탕으로 예측
- ✓ count(빈도)를 기반으로 확률 계산
- ✓ sparsity problem (n-gram chunk 문서 내 등장 x)

Neural Language Model



A neural probabilistic language model (Bengio et al, 2003)

- ✓ 미리 지정한 window size 이전 단어를 바탕으로 예측
- ✓ word embedding: 단어의 distributed representation 학습
- ✓ window size가 커지면 파라미터 수 증가 -> 연산 부담, 과적합 가능

01 Language Model

N-gram LM, Neural LM

N-gram Language Model

$$P(\text{코딩하기}, \text{좋은}, \text{카페}, \text{추천})$$

uni-gram

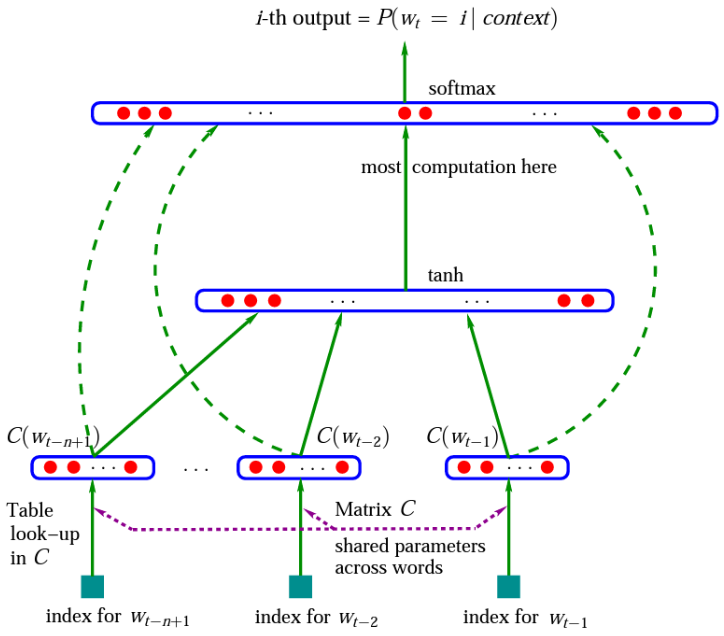
$P(\text{코딩하기})$ $P(\text{좋은})$ $P(\text{카페})$ $P(\text{추천})$

bi-gram

$P(\text{코딩하기})$ $P(\text{좋은}|\ \text{코딩하기})$ $P(\text{카페}|\ \text{좋은})$ $P(\text{추천}|\ \text{카페})$

$$P(\text{카페}|\ \text{코딩하기}, \text{좋은}) = \frac{count(\text{코딩하기}, \text{좋은}, \text{카페})}{count(\text{코딩하기}, \text{좋은})}$$

Neural Language Model



A neural probabilistic language model (Bengio et al, 2003)

✓ 입력 길이가 고정되어(N, window size) 이전에 등장하는 모든 단어를 고려할 수 없음

02 RNN Language Model

Architecture

④ Output distribution

$\hat{\boldsymbol{y}}^{(t)} = \mathrm{softmax}\left(\boldsymbol{U}\boldsymbol{h}^{(t)} + \boldsymbol{b}_2\right) \in \mathbb{R}^{|V|}$

③ Hidden States

$\boldsymbol{h}^{(t)} = \sigma\left(\boldsymbol{W}_h\boldsymbol{h}^{(t-1)} + \boldsymbol{W}_e\boldsymbol{e}^{(t)} + \boldsymbol{b}_1\right)$

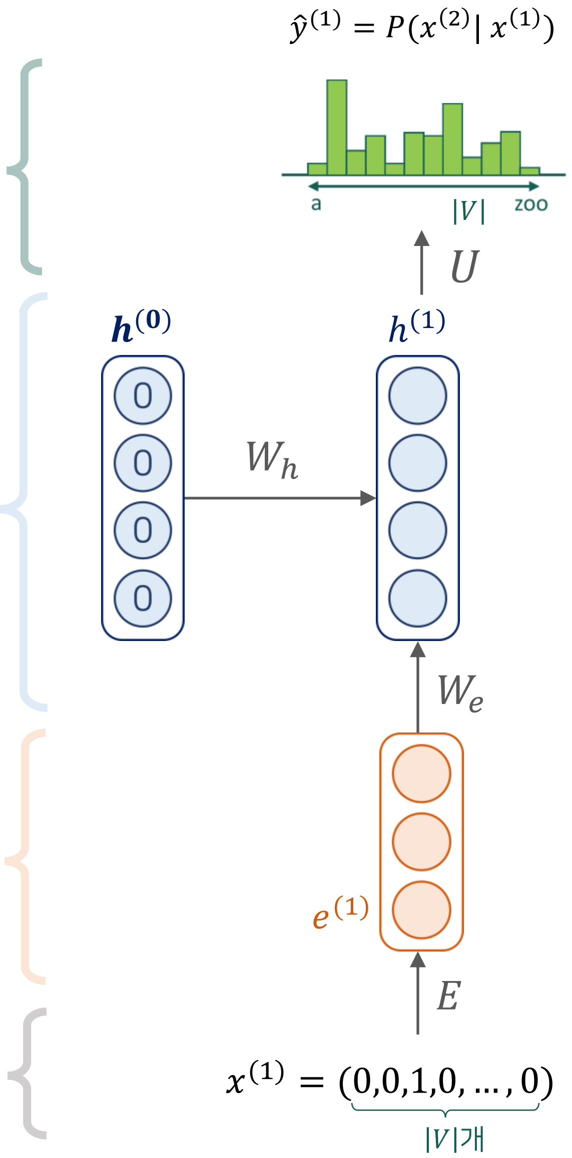
$\boldsymbol{W}_{\boldsymbol{h}}: (d_h \times d_h)$, $\boldsymbol{W}_{\boldsymbol{e}}: (d_e \times d_h)$

② Word embedding

$\boldsymbol{e}^{(t)} = \boldsymbol{E}\boldsymbol{x}^{(t)}$ $\quad \boldsymbol{E}: (|V| \times d_e)$

① Input word sequence

- 원핫 벡터 $\boldsymbol{x}^{(t)} \in \mathbb{R}^{|V|}$



02 RNN Language Model

Architecture

④ Output distribution

$\hat{\boldsymbol{y}}^{(t)} = \mathrm{softmax}\left(\boldsymbol{U}\boldsymbol{h}^{(t)} + \boldsymbol{b}_2\right) \in \mathbb{R}^{|V|}$

③ Hidden States

$\boldsymbol{h}^{(t)} = \sigma\left(\boldsymbol{W}_h\boldsymbol{h}^{(t-1)} + \boldsymbol{W}_e\boldsymbol{e}^{(t)} + \boldsymbol{b}_1\right)$

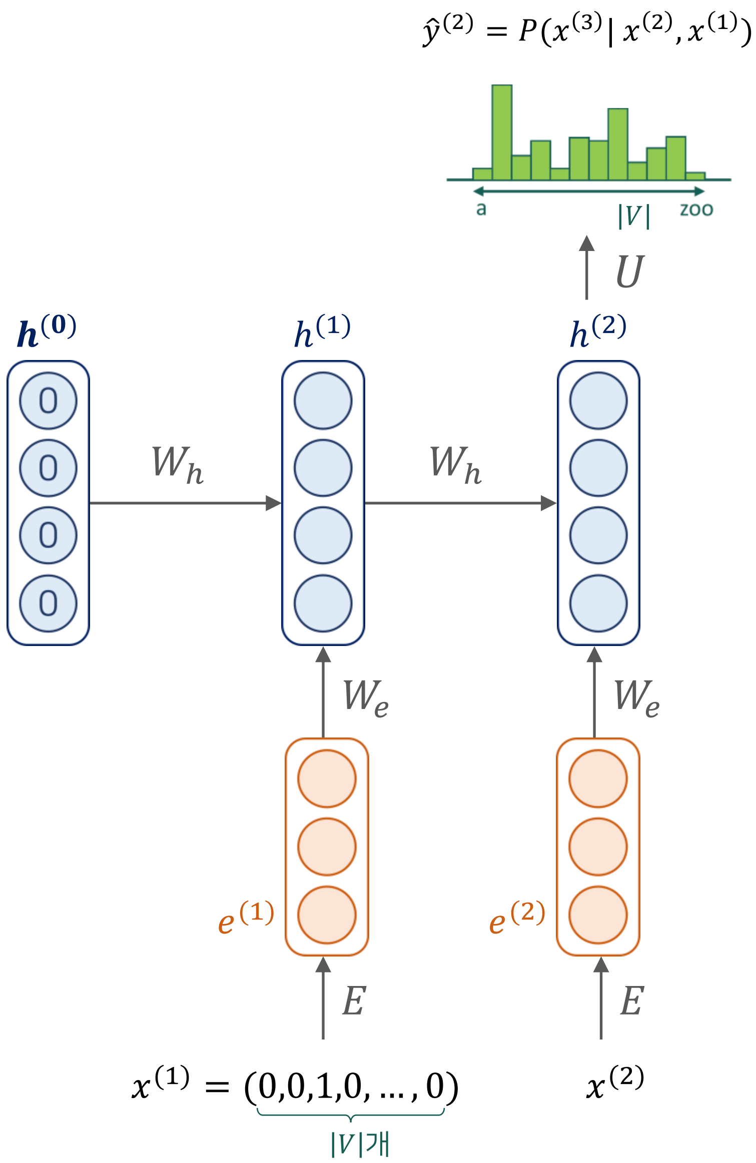
$\boldsymbol{W_h}: (d_h \times d_h),\ \boldsymbol{W_e}: (d_e \times d_h)$

② Word embedding

$\boldsymbol{e}^{(t)} = \boldsymbol{E}\boldsymbol{x}^{(t)} \quad \boldsymbol{E}: (|V| \times d_e)$

① Input word sequence

- 원핫 벡터 $\boldsymbol{x}^{(t)} \in \mathbb{R}^{|V|}$



02 RNN Language Model

Architecture

④ Output distribution

$\hat{\boldsymbol{y}}^{(t)} = \mathrm{softmax}\left(\boldsymbol{U}\boldsymbol{h}^{(t)} + \boldsymbol{b}_2\right) \in \mathbb{R}^{|V|}$

③ Hidden States

$\boldsymbol{h}^{(t)} = \sigma\left(\boldsymbol{W}_h\boldsymbol{h}^{(t-1)} + \boldsymbol{W}_e\boldsymbol{e}^{(t)} + \boldsymbol{b}_1\right)$

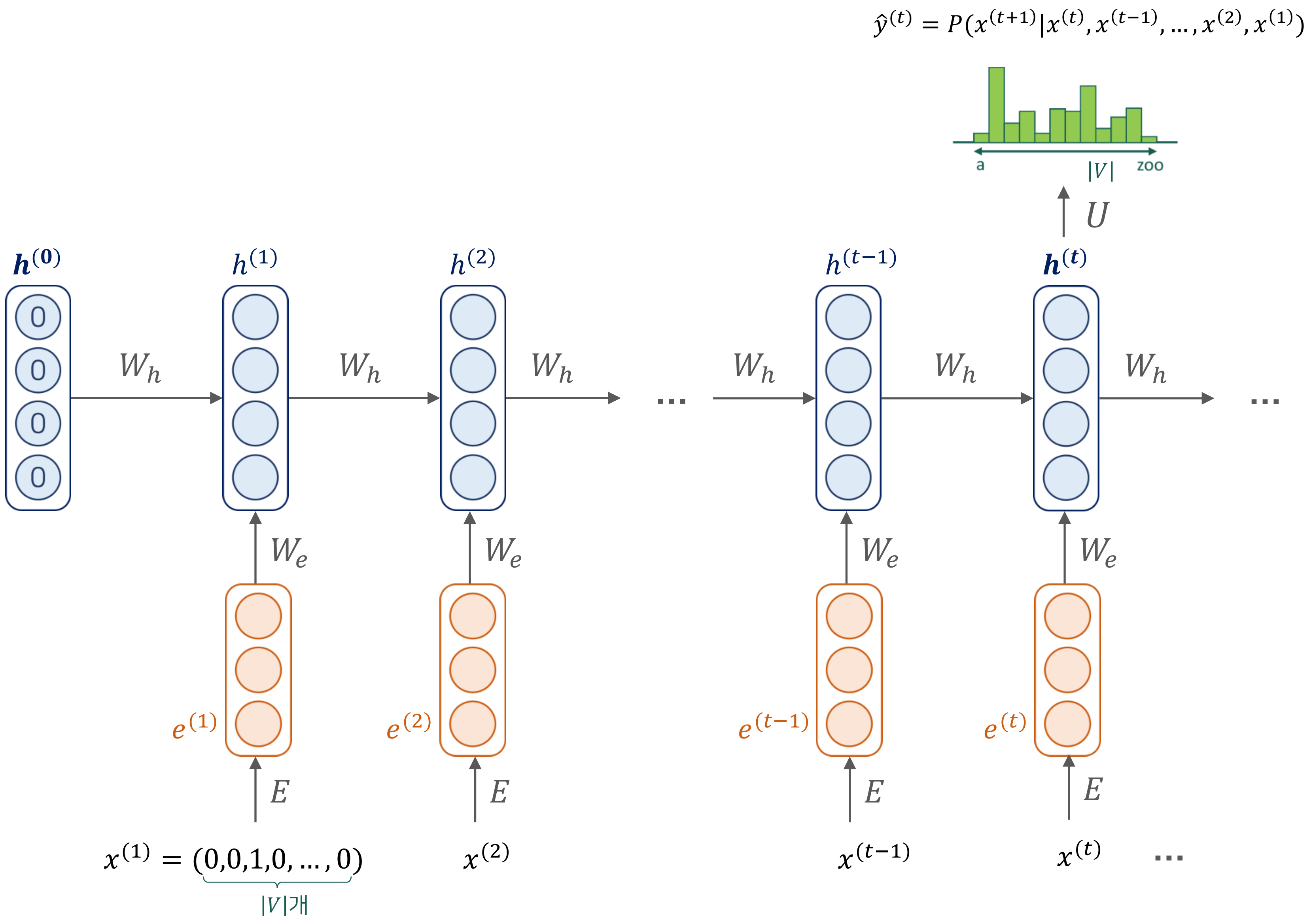
$\boldsymbol{W_h}: (d_h \times d_h),\ \boldsymbol{W_e}: (d_e \times d_h)$

② Word embedding

$\boldsymbol{e}^{(t)} = \boldsymbol{E}\boldsymbol{x}^{(t)} \qquad \boldsymbol{E}: (|V| \times d_e)$

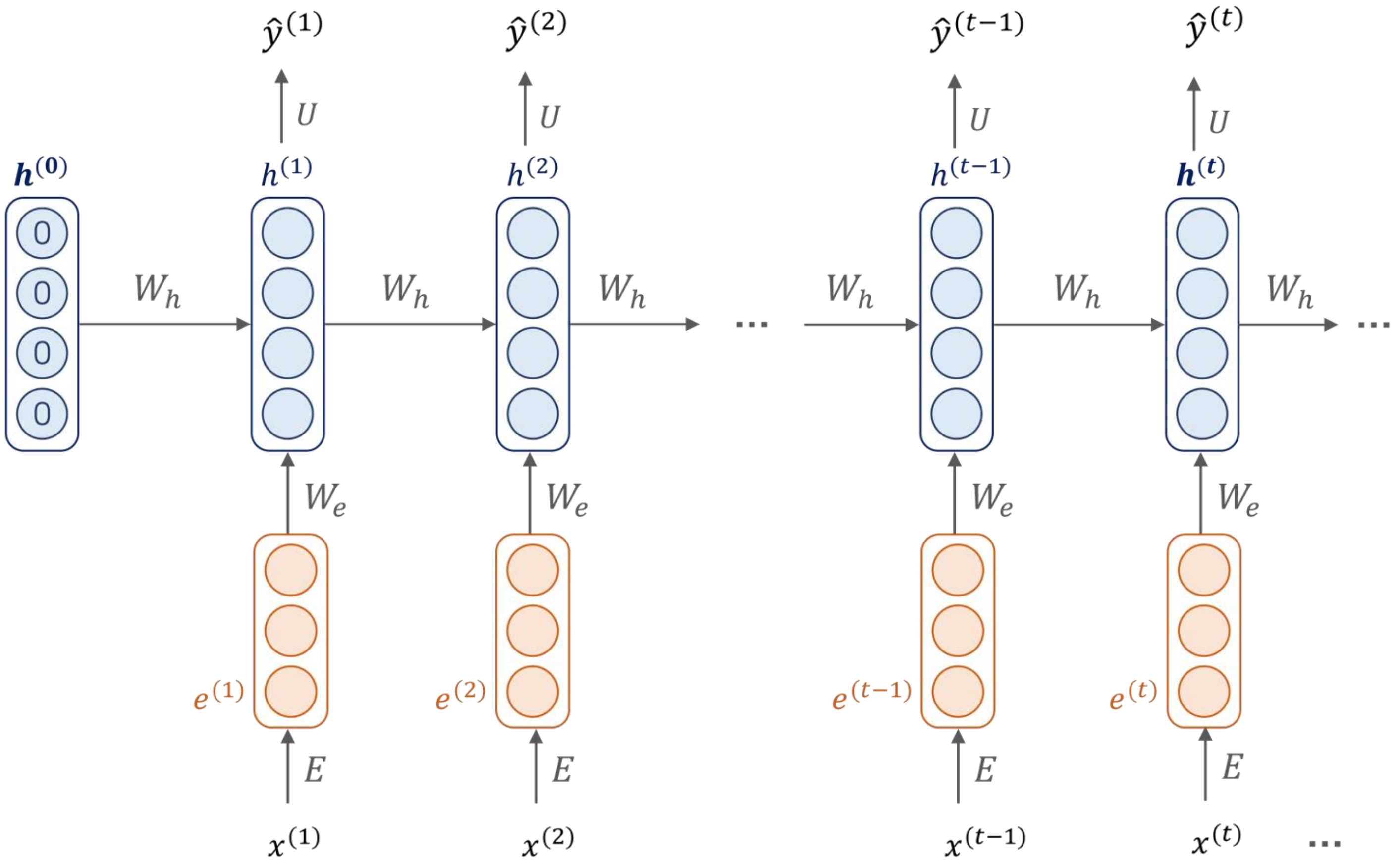
① Input word sequence

- 원핫 벡터 $\boldsymbol{x}^{(t)} \in \mathbb{R}^{|V|}$



02 RNN Language Model

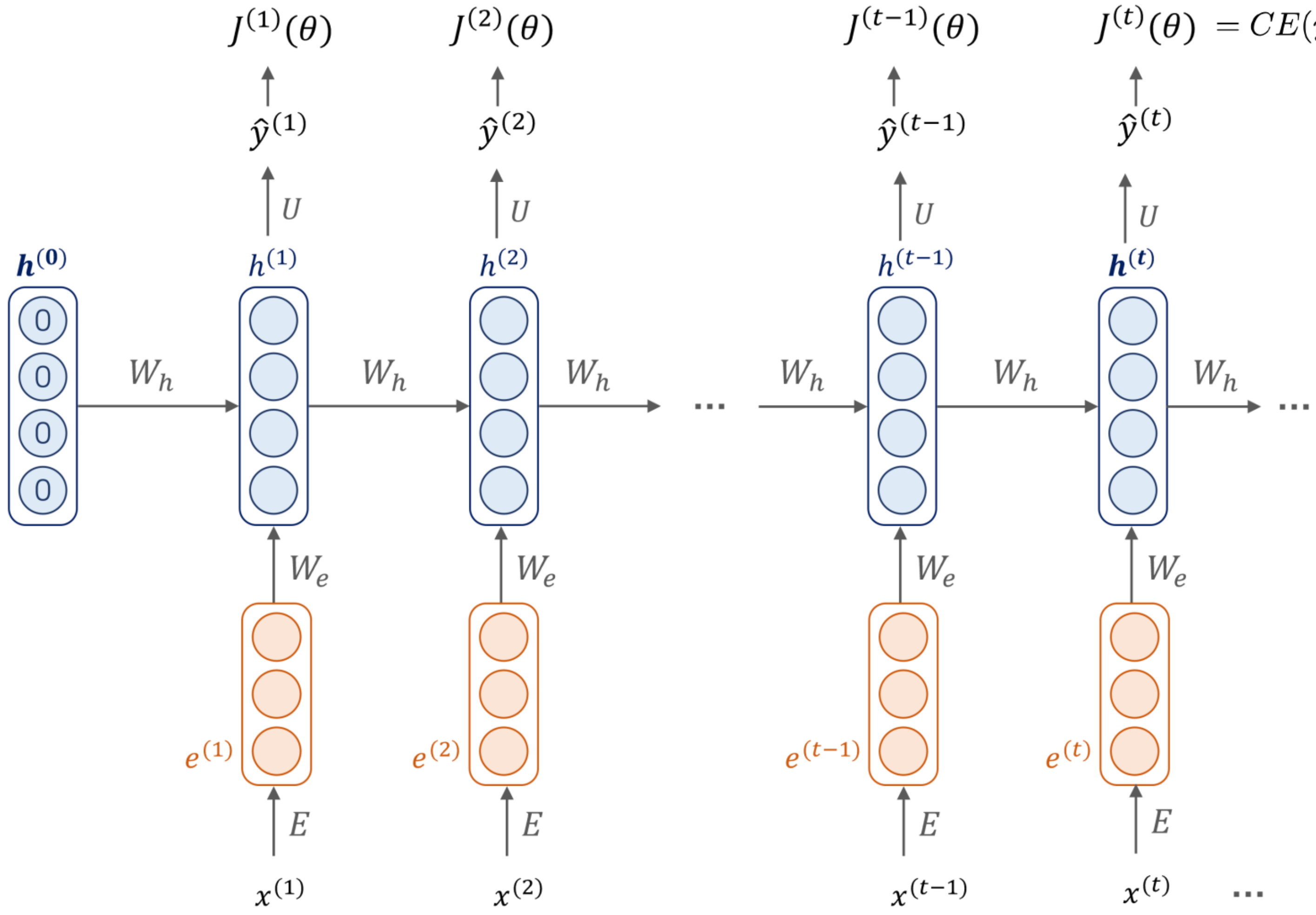
Loss Function



- ✓ many-to-many
- ✓ 매 time step마다 입,출력 존재

02 RNN Language Model

Loss Function



$$J^{(t)}(\theta) = CE(\boldsymbol{y}^{(t)}, \hat{\boldsymbol{y}}^{(t)}) = -\sum_{w \in V} \boldsymbol{y}_w^{(t)} \log \hat{\boldsymbol{y}}_w^{(t)} = -\log \hat{\boldsymbol{y}}_{\boldsymbol{x}_{t+1}}^{(t)}$$

$$J(\theta) = \frac{1}{T}\sum_{t=1}^{T} J^{(t)}(\theta) = \frac{1}{T}\sum_{t=1}^{T} -\log \hat{\boldsymbol{y}}_{\boldsymbol{x}_{t+1}}^{(t)}$$

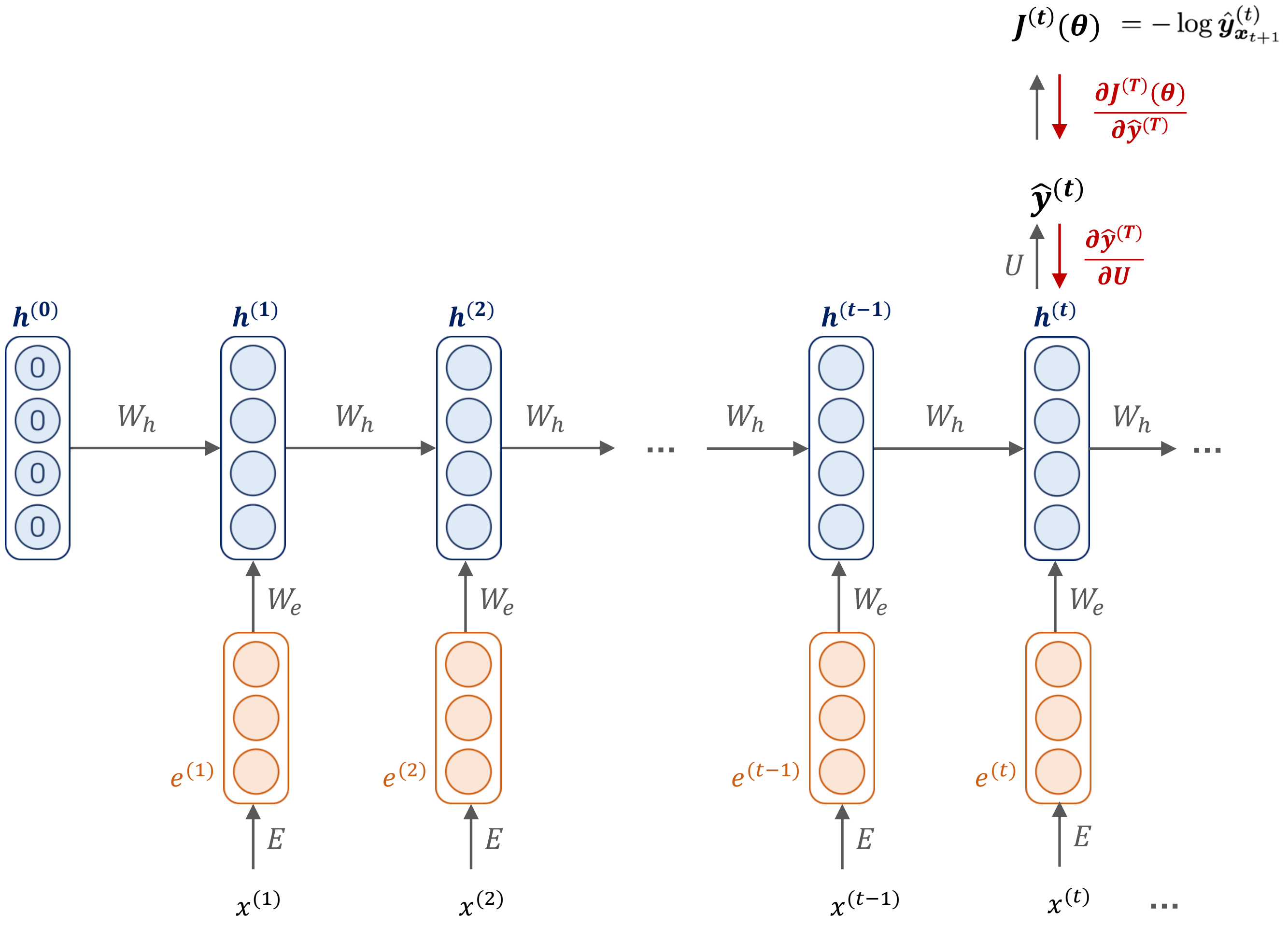
- ✓ many-to-many
- ✓ 매 time step마다 입,출력 존재

02 RNN Language Model

Back Propagation Through Time (BPTT) ①

$$\hat{\boldsymbol{y}}^{(t)} = \text{softmax}\left(\boxed{\boldsymbol{U}}\boldsymbol{h}^{(t)} + \boldsymbol{b}_2\right) \in \mathbb{R}^{|V|}$$

$$\boldsymbol{h}^{(t)} = \sigma\left(\boldsymbol{W}_h \boldsymbol{h}^{(t-1)} + \boldsymbol{W}_e \boldsymbol{e}^{(t)} + \boldsymbol{b}_1\right)$$



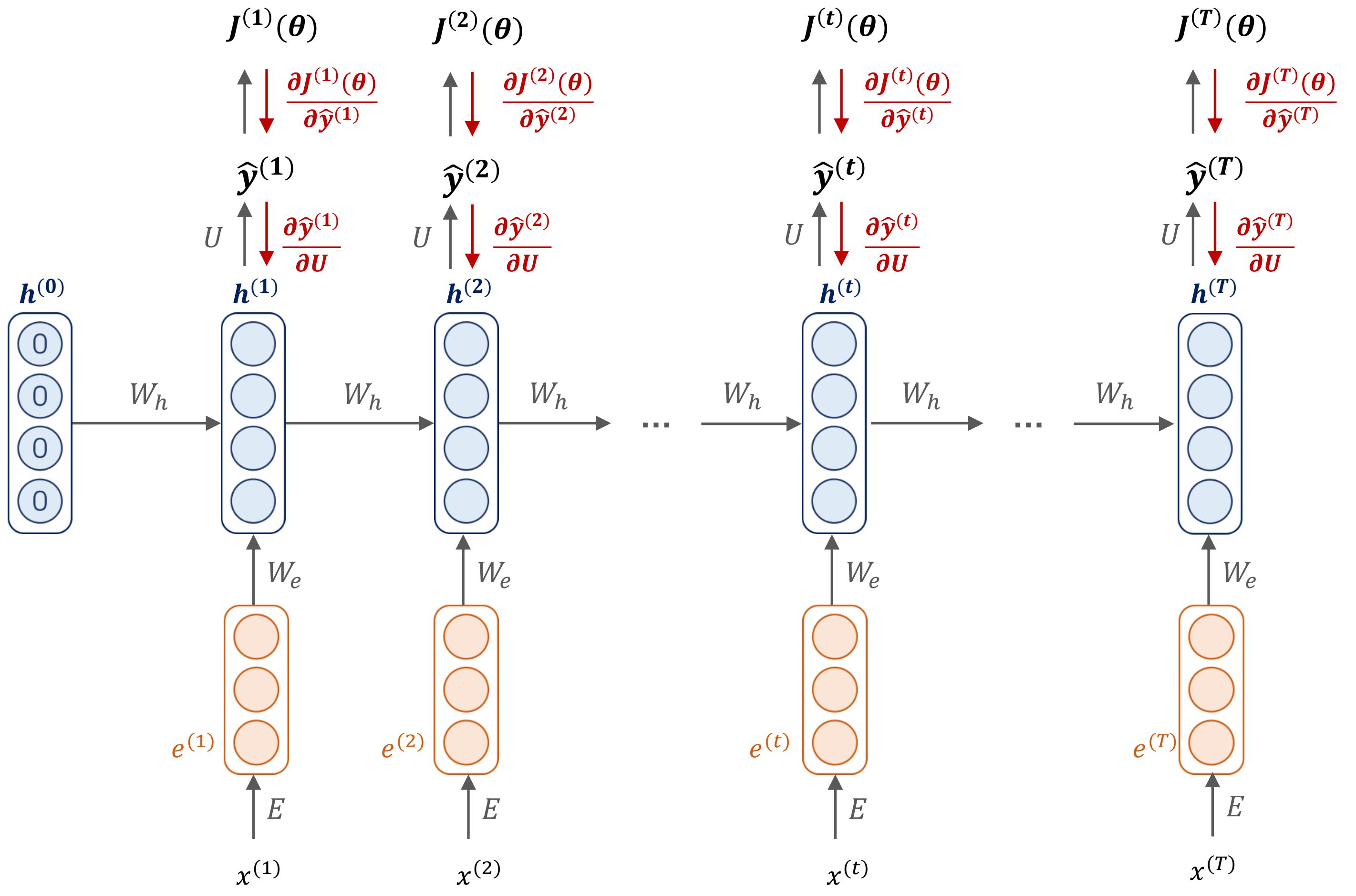
$$① \quad \frac{\partial J^{(t)}(\theta)}{\partial U} = \frac{\partial J^{(t)}(\theta)}{\partial \hat{y}^{(t)}} \cdot \frac{\partial \hat{y}^{(t)}}{\partial U}$$

02 RNN Language Model

Back Propagation Through Time (BPTT) ①

$$\hat{\boldsymbol{y}}^{(t)} = \mathrm{softmax}\left(\boxed{\boldsymbol{U}}\boldsymbol{h}^{(t)} + \boldsymbol{b}_2\right) \in \mathbb{R}^{|V|}$$

$$\boldsymbol{h}^{(t)} = \sigma\left(\boldsymbol{W}_h \boldsymbol{h}^{(t-1)} + \boldsymbol{W}_e \boldsymbol{e}^{(t)} + \boldsymbol{b}_1\right)$$



① $\frac{\partial J^{(t)}(\theta)}{\partial U} = \frac{\partial J^{(t)}(\theta)}{\partial \hat{y}^{(t)}} \cdot \frac{\partial \hat{y}^{(t)}}{\partial U}$

$$\frac{\partial J(\theta)}{\partial U} = \sum_{t=1}^{T} \frac{\partial J^{(t)}(\theta)}{\partial U}$$

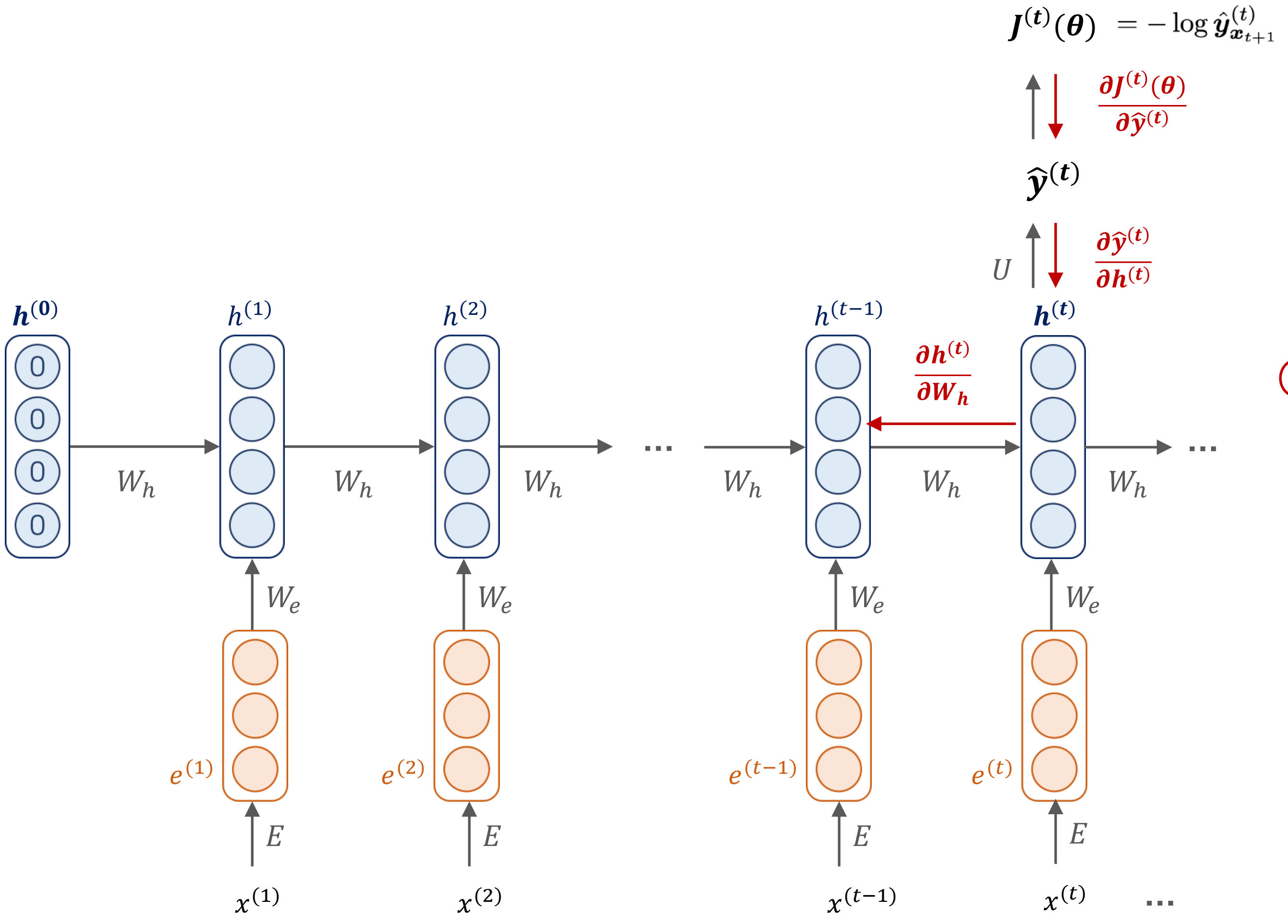
$$U^{new} = U^{old} - \alpha \frac{\partial J(\theta)}{\partial U}$$

02 RNN Language Model

Back Propagation Through Time (BPTT) ②

$$\hat{\boldsymbol{y}}^{(t)} = \mathrm{softmax}\left(\boldsymbol{U}\boldsymbol{h}^{(t)} + \boldsymbol{b}_2\right) \in \mathbb{R}^{|V|}$$

$$\boldsymbol{h}^{(t)} = \sigma\left(\boxed{\boldsymbol{W}_h}\boldsymbol{h}^{(t-1)} + \boldsymbol{W}_e\boldsymbol{e}^{(t)} + \boldsymbol{b}_1\right)$$



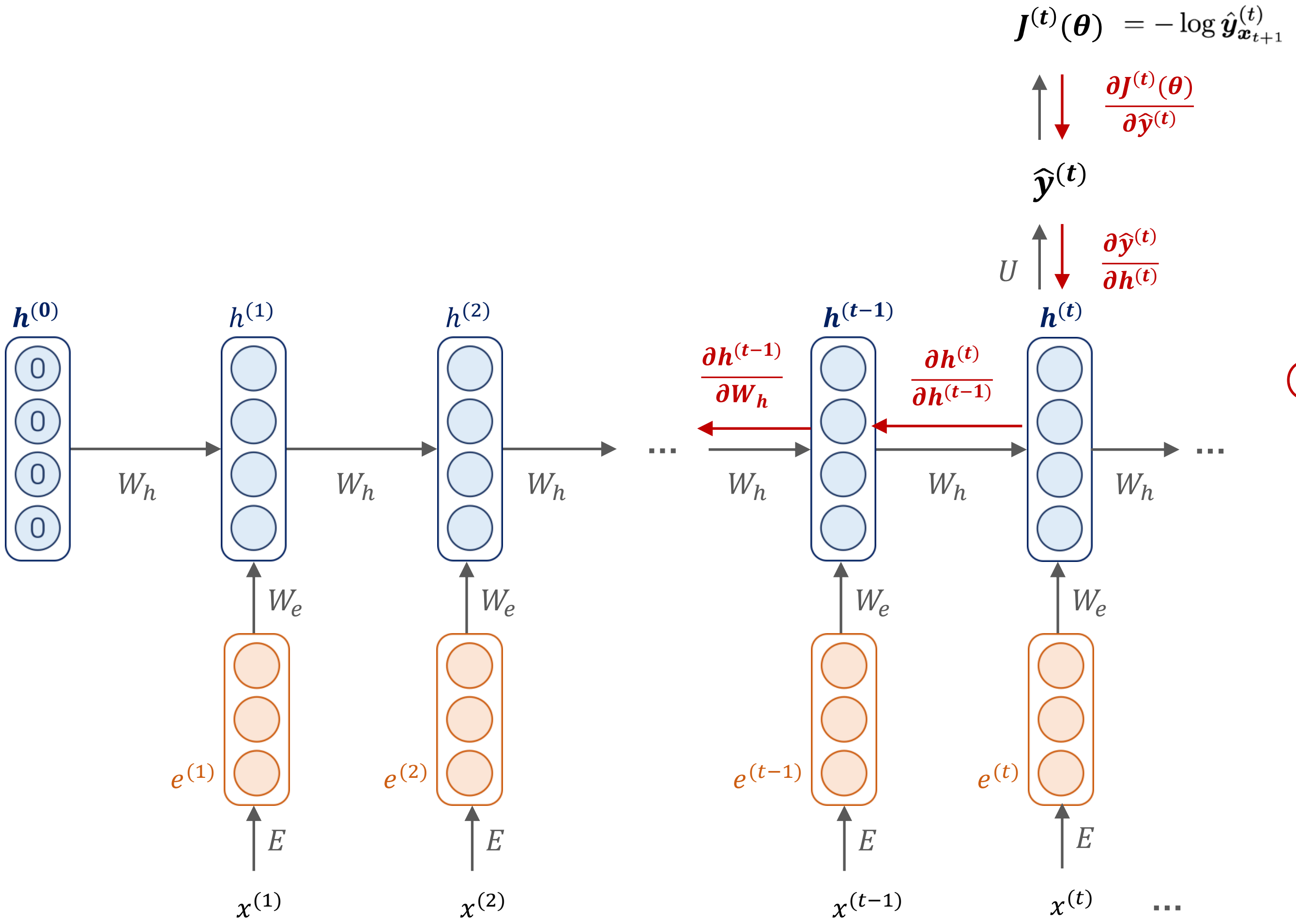
② $$\frac{\partial J^{(t)}(\theta)}{\partial W_h} = \frac{\partial J^{(t)}(\theta)}{\partial \hat{y}^{(t)}} \cdot \frac{\partial \hat{y}^{(t)}}{\partial h^{(t)}} \cdot \frac{\partial h^{(t)}}{\partial W_h}$$

02 RNN Language Model

Back Propagation Through Time (BPTT) ②

$$\hat{\boldsymbol{y}}^{(t)} = \mathrm{softmax}\left(\boldsymbol{U}\boldsymbol{h}^{(t)} + \boldsymbol{b}_2\right) \in \mathbb{R}^{|V|}$$

$$\boldsymbol{h}^{(t)} = \sigma\left(\boxed{\boldsymbol{W}_h}\boldsymbol{h}^{(t-1)} + \boldsymbol{W}_e\boldsymbol{e}^{(t)} + \boldsymbol{b}_1\right)$$



$$② \ \frac{\partial J^{(t)}(\theta)}{\partial W_h} = \frac{\partial J^{(t)}(\theta)}{\partial \hat{y}^{(t)}} \cdot \frac{\partial \hat{y}^{(t)}}{\partial h^{(t)}} \cdot \frac{\partial h^{(t)}}{\partial W_h}$$

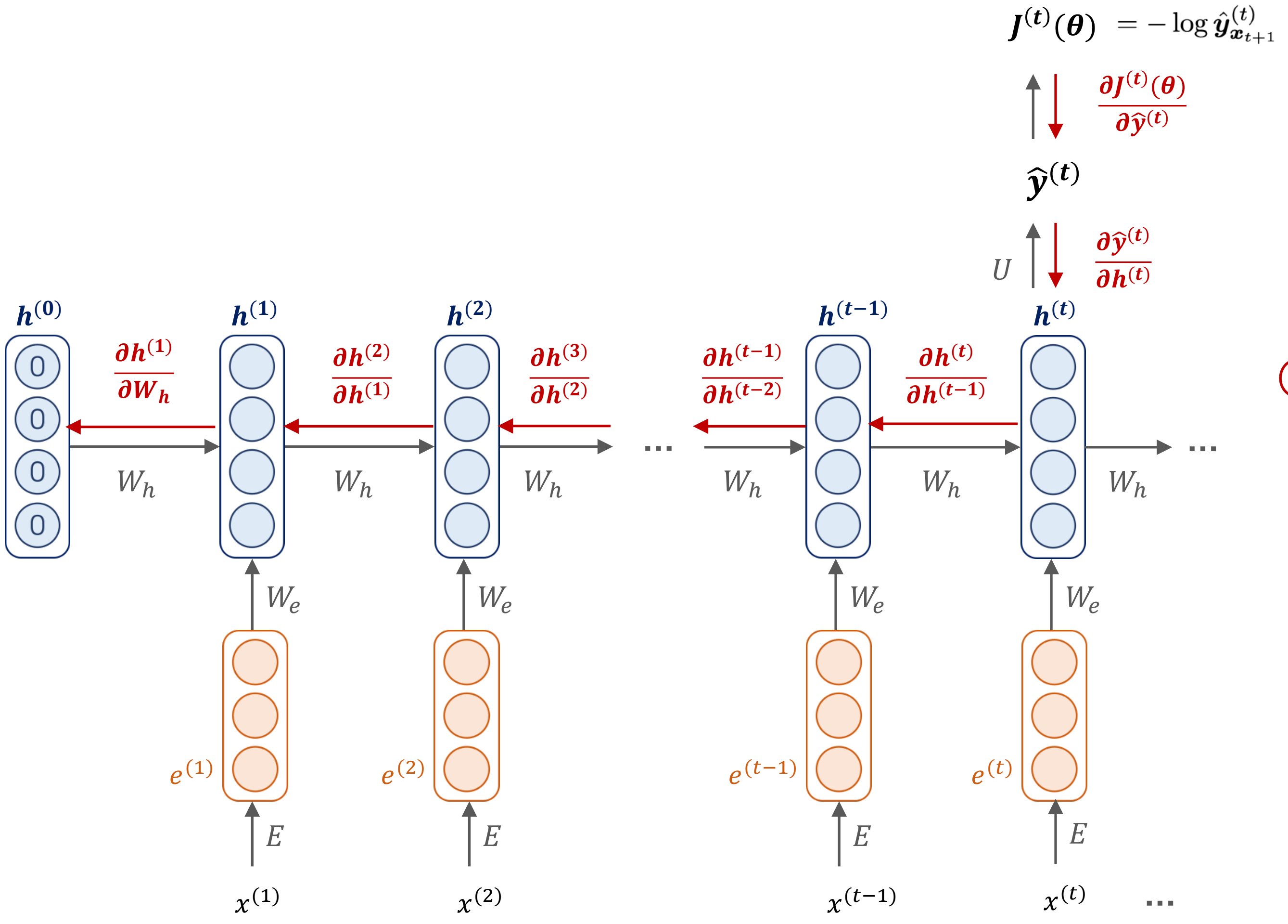
$$+ \frac{\partial J^{(t)}(\theta)}{\partial \hat{y}^{(t)}} \cdot \frac{\partial \hat{y}^{(t)}}{\partial h^{(t)}} \cdot \frac{\partial h^{(t)}}{\partial h^{(t-1)}} \cdot \frac{\partial h^{(t-1)}}{\partial W_h}$$

02 RNN Language Model

Back Propagation Through Time (BPTT) ②

$$\hat{\boldsymbol{y}}^{(t)} = \mathrm{softmax}\left(\boldsymbol{U}\boldsymbol{h}^{(t)} + \boldsymbol{b}_2\right) \in \mathbb{R}^{|V|}$$

$$\boldsymbol{h}^{(t)} = \sigma\left(\boxed{\boldsymbol{W}_h}\boldsymbol{h}^{(t-1)} + \boldsymbol{W}_e\boldsymbol{e}^{(t)} + \boldsymbol{b}_1\right)$$



$$② \ \frac{\partial J^{(t)}(\theta)}{\partial W_h} = \frac{\partial J^{(t)}(\theta)}{\partial \hat{y}^{(t)}} \cdot \frac{\partial \hat{y}^{(t)}}{\partial h^{(t)}} \cdot \frac{\partial h^{(t)}}{\partial W_h}$$

$$+ \frac{\partial J^{(t)}(\theta)}{\partial \hat{y}^{(t)}} \cdot \frac{\partial \hat{y}^{(t)}}{\partial h^{(t)}} \cdot \frac{\partial h^{(t)}}{\partial h^{(t-1)}} \cdot \frac{\partial h^{(t-1)}}{\partial W_h}$$

$$\vdots$$

$$+ \frac{\partial J^{(t)}(\theta)}{\partial \hat{y}^{(t)}} \cdot \frac{\partial \hat{y}^{(t)}}{\partial h^{(t)}} \cdot \frac{\partial h^{(t)}}{\partial h^{(t-1)}} \cdot \cdots \cdot \frac{\partial h^{(2)}}{\partial h^{(1)}} \cdot \frac{\partial h^{(1)}}{\partial W_h}$$

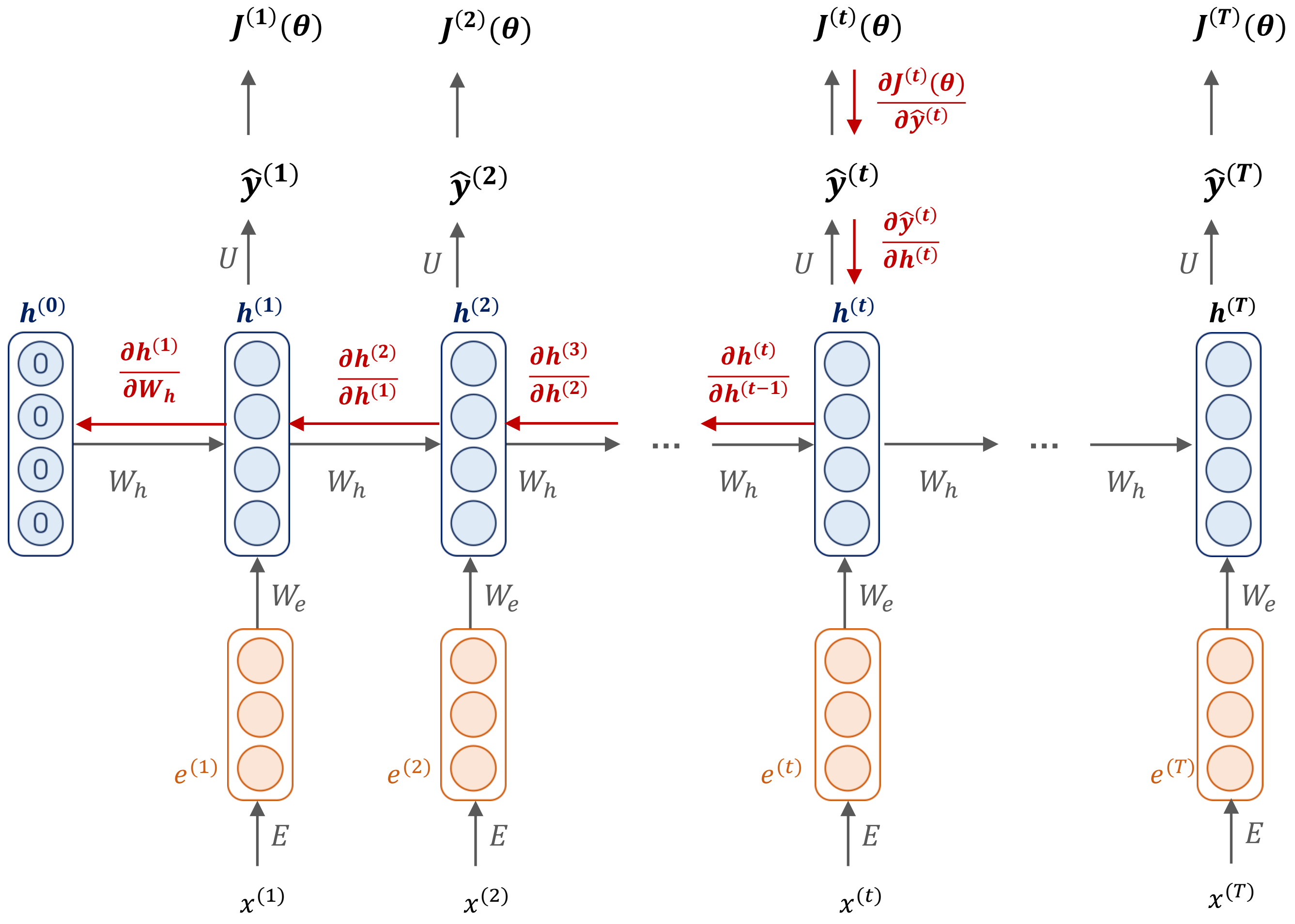
$$= \sum_{i=1}^{t} \frac{\partial J^{(t)}(\theta)}{\partial \hat{y}^{(t)}} \cdot \frac{\partial \hat{y}^{(t)}}{\partial h^{(t)}} \cdot \frac{\partial h^{(t)}}{\partial h^{(i)}} \cdot \frac{\partial h^{(i)}}{\partial W_h}$$

02 RNN Language Model

Back Propagation Through Time (BPTT) ②

$$\hat{\boldsymbol{y}}^{(t)} = \text{softmax}\left(\boldsymbol{U}\boldsymbol{h}^{(t)} + \boldsymbol{b}_2\right) \in \mathbb{R}^{|V|}$$

$$\boldsymbol{h}^{(t)} = \sigma\left(\boxed{\boldsymbol{W}_h}\boldsymbol{h}^{(t-1)} + \boldsymbol{W}_e\boldsymbol{e}^{(t)} + \boldsymbol{b}_1\right)$$



② $$\frac{\partial J^{(t)}(\theta)}{\partial W_h} = \sum_{i=1}^{t} \frac{\partial J^{(t)}(\theta)}{\partial \hat{y}^{(t)}} \cdot \frac{\partial \hat{y}^{(t)}}{\partial h^{(t)}} \cdot \frac{\partial h^{(t)}}{\partial h^{(i)}} \cdot \frac{\partial h^{(i)}}{\partial W_h}$$

$$\Rightarrow \quad \frac{\partial J(\theta)}{\partial W_h} =$$

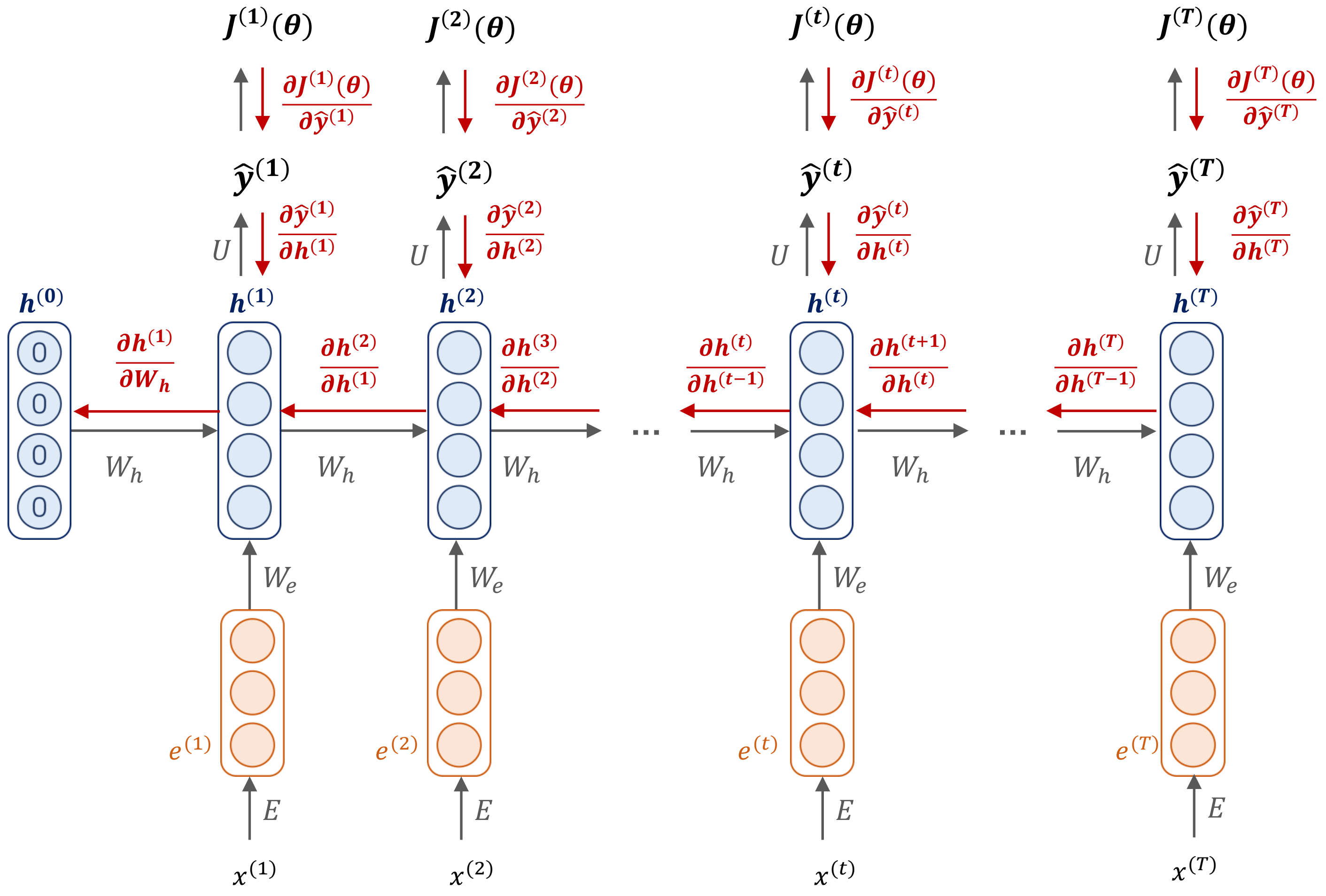
02

RNN Language Model

Back Propagation Through Time (BPTT) ②

$$\hat{\boldsymbol{y}}^{(t)} = \mathrm{softmax}\left(\boldsymbol{U}\boldsymbol{h}^{(t)} + \boldsymbol{b}_2\right) \in \mathbb{R}^{|V|}$$

$$\boldsymbol{h}^{(t)} = \sigma\left(\boxed{\boldsymbol{W}_h}\boldsymbol{h}^{(t-1)} + \boldsymbol{W}_e\boldsymbol{e}^{(t)} + \boldsymbol{b}_1\right)$$



$$② \quad \frac{\partial J^{(t)}(\theta)}{\partial W_h} = \sum_{i=1}^{t} \frac{\partial J^{(t)}(\theta)}{\partial \hat{y}^{(t)}} \cdot \frac{\partial \hat{y}^{(t)}}{\partial h^{(t)}} \cdot \frac{\partial h^{(t)}}{\partial h^{(i)}} \cdot \frac{\partial h^{(i)}}{\partial W_h}$$

$$\Rightarrow \quad \frac{\partial J(\theta)}{\partial W_h} = \sum_{t=1}^{T} \frac{\partial J^{(t)}(\theta)}{\partial W_h}$$

$$W_h^{\,new} = W_h^{\,old} - \alpha \frac{\partial J(\theta)}{\partial W_h}$$

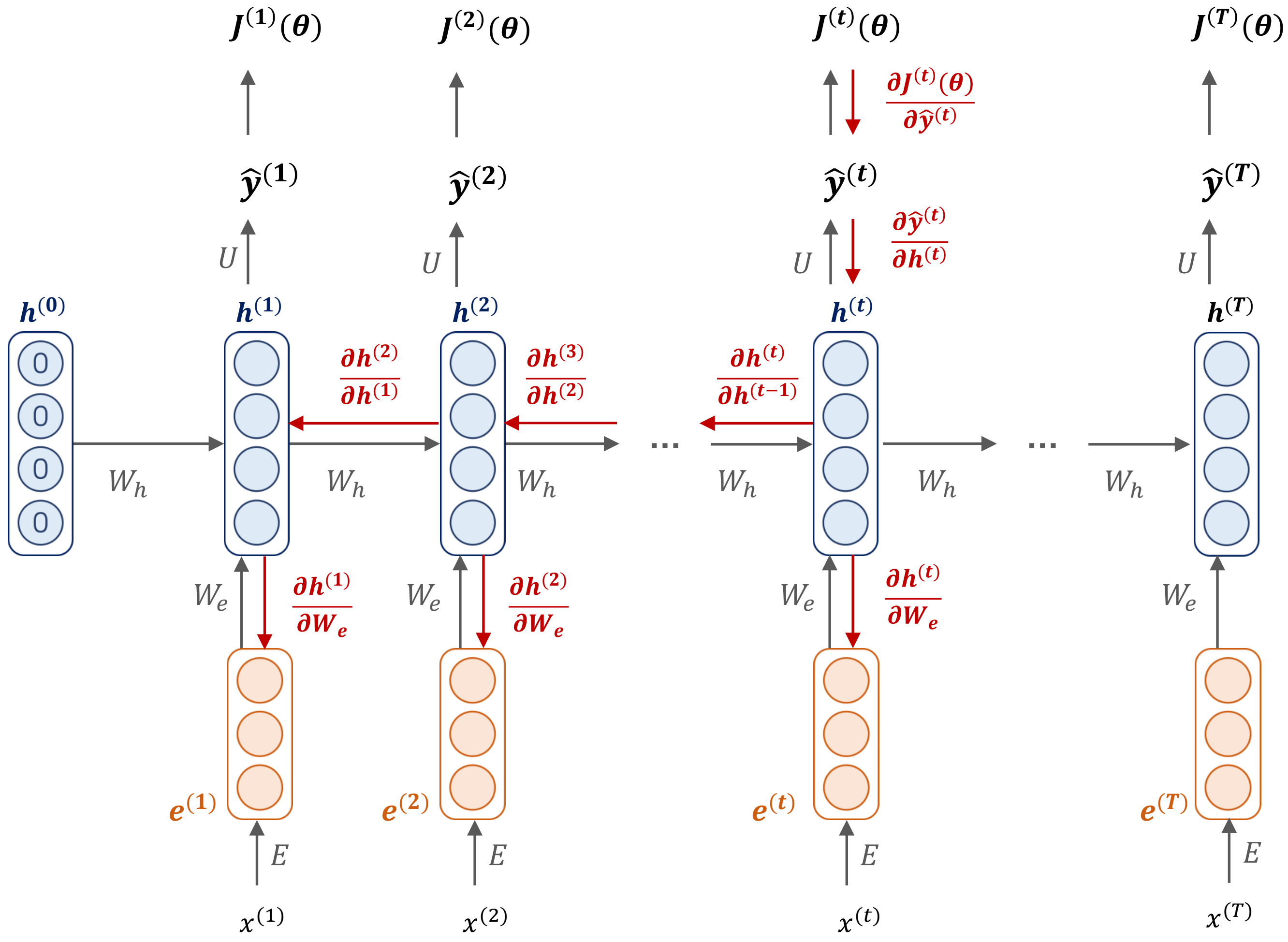
02

RNN Language Model

Back Propagation Through Time (BPTT) ③

$$\hat{\boldsymbol{y}}^{(t)} = \mathrm{softmax}\left(\boldsymbol{U}\boldsymbol{h}^{(t)} + \boldsymbol{b}_2\right) \in \mathbb{R}^{|V|}$$

$$\boldsymbol{h}^{(t)} = \sigma\left(\boldsymbol{W}_h\boldsymbol{h}^{(t-1)} + \boxed{\boldsymbol{W}_e}\boldsymbol{e}^{(t)} + \boldsymbol{b}_1\right)$$



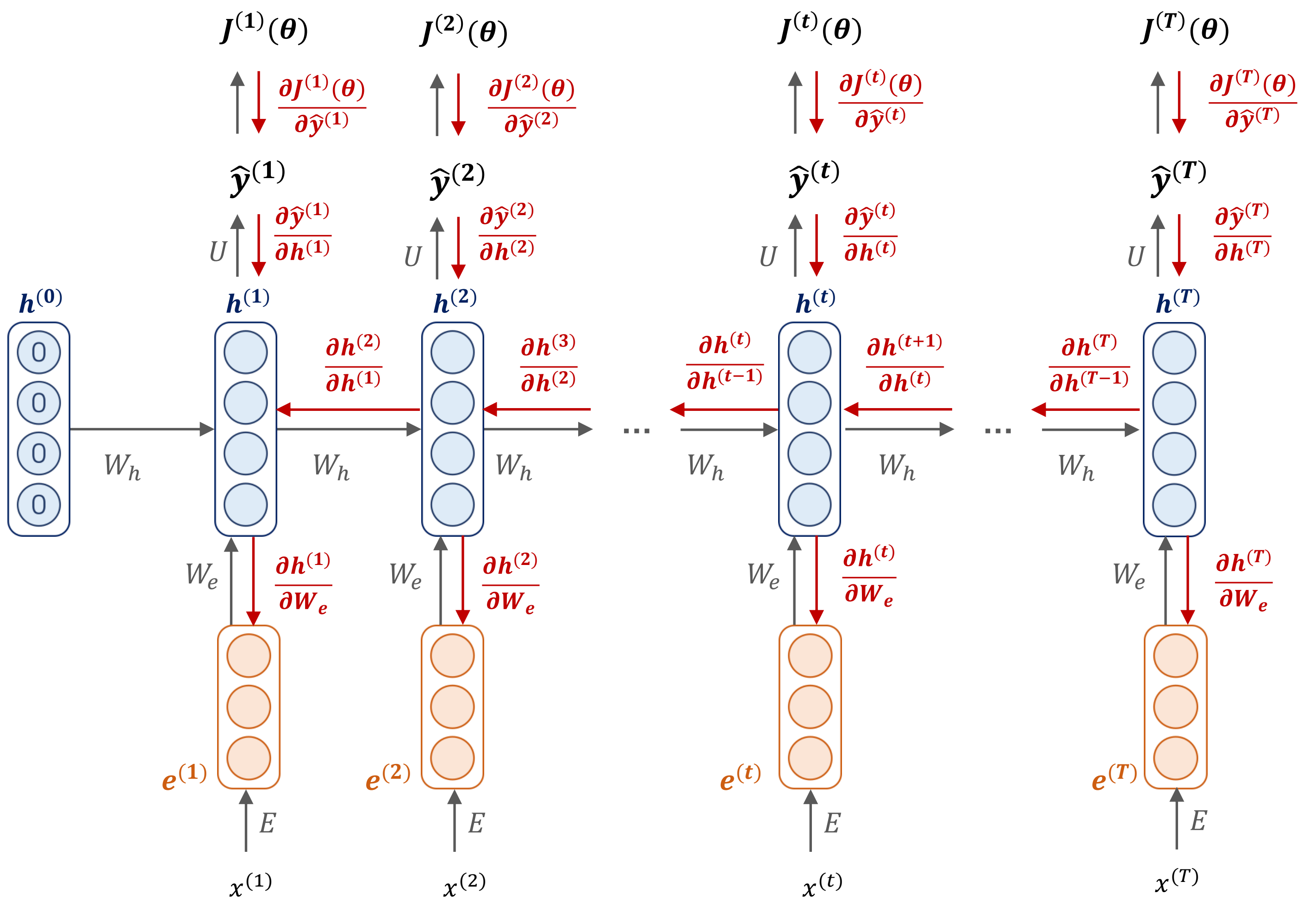
③ $$\frac{\partial J^{(t)}(\theta)}{\partial W_e} = \sum_{i=1}^{t} \frac{\partial J^{(t)}(\theta)}{\partial \hat{y}^{(t)}} \cdot \frac{\partial \hat{y}^{(t)}}{\partial h^{(t)}} \cdot \frac{\partial h^{(t)}}{\partial h^{(i)}} \cdot \frac{\partial h^{(i)}}{\partial W_e}$$

02 RNN Language Model

Back Propagation Through Time (BPTT) ③

$$\hat{\boldsymbol{y}}^{(t)} = \mathrm{softmax}\left(\boldsymbol{U}\boldsymbol{h}^{(t)} + \boldsymbol{b}_2\right) \in \mathbb{R}^{|V|}$$

$$\boldsymbol{h}^{(t)} = \sigma\left(\boldsymbol{W}_h\boldsymbol{h}^{(t-1)} + \boxed{\boldsymbol{W}_e}\boldsymbol{e}^{(t)} + \boldsymbol{b}_1\right)$$



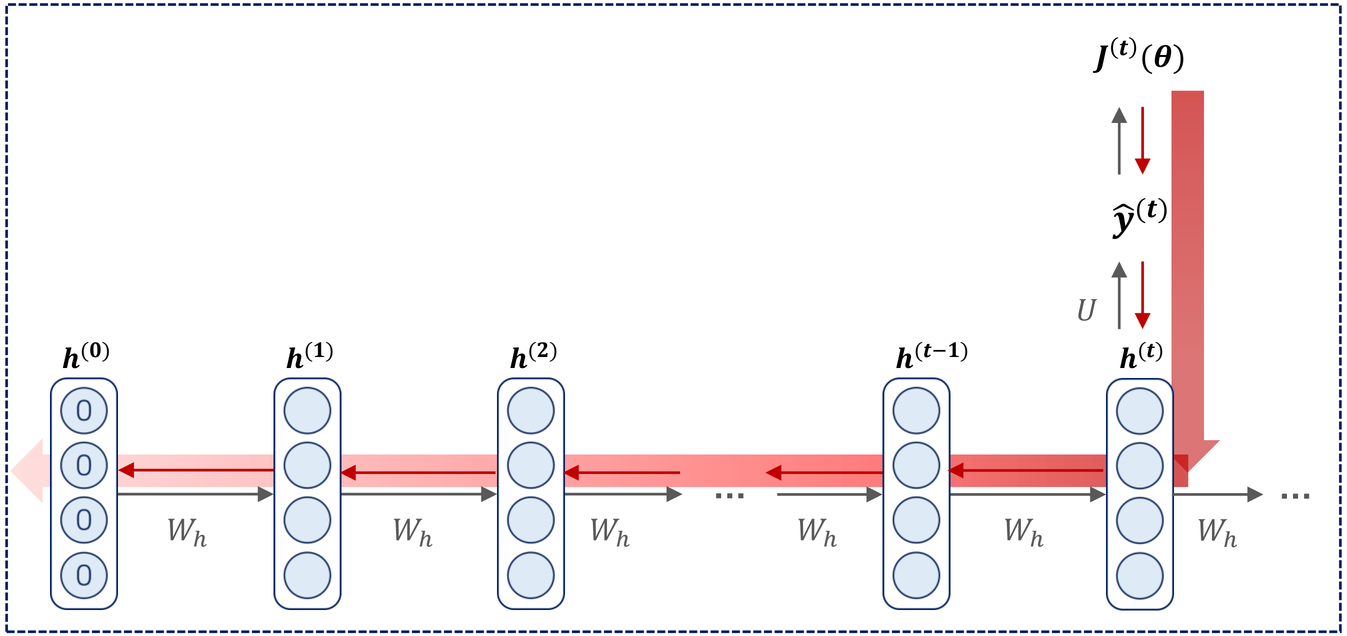
③ $$\frac{\partial J^{(t)}(\theta)}{\partial W_e} = \sum_{i=1}^{t} \frac{\partial J^{(t)}(\theta)}{\partial \hat{y}^{(t)}} \cdot \frac{\partial \hat{y}^{(t)}}{\partial h^{(t)}} \cdot \frac{\partial h^{(t)}}{\partial h^{(i)}} \cdot \frac{\partial h^{(i)}}{\partial W_e}$$

$$\Rightarrow \quad \frac{\partial J(\theta)}{\partial W_e} = \sum_{t=1}^{T} \frac{\partial J^{(t)}(\theta)}{\partial W_e}$$

$$W_e^{\,new} = W_e^{\,old} - \alpha \frac{\partial J(\theta)}{\partial W_e}$$

03 Problem of RNN

① Vanishing/Exploding Gradients



$$\hat{\boldsymbol{y}}^{(t)} = \mathrm{softmax}\left(\boldsymbol{U}\boldsymbol{h}^{(t)} + \boldsymbol{b}_2\right) \in \mathbb{R}^{|V|}$$

$$\boldsymbol{h}^{(t)} = \sigma\left(\underbrace{\boxed{\boldsymbol{W}_h}\boldsymbol{h}^{(t-1)} + \boldsymbol{W}_e\boldsymbol{e}^{(t)} + \boldsymbol{b}_1}_{=\boldsymbol{s}^{(t)}}\right)$$

$$\frac{\partial \boldsymbol{h}^{(t)}}{\partial \boldsymbol{h}^{(t-1)}} = \frac{\partial \boldsymbol{h}^{(t)}}{\partial \boldsymbol{s}^{(t)}} \cdot \frac{\partial \boldsymbol{s}^{(t)}}{\partial \boldsymbol{h}^{(t-1)}} = \frac{\partial \boldsymbol{h}^{(t)}}{\partial \boldsymbol{s}^{(t)}} \cdot \boldsymbol{W}_h$$

$$\frac{\partial J^{(t)}(\theta)}{\partial W_h} = \frac{\partial J^{(t)}(\theta)}{\partial \hat{y}^{(t)}} \cdot \frac{\partial \hat{y}^{(t)}}{\partial h^{(t)}} \cdot \frac{\partial h^{(t)}}{\partial W_h} + \frac{\partial J^{(t)}(\theta)}{\partial \hat{y}^{(t)}} \cdot \frac{\partial \hat{y}^{(t)}}{\partial h^{(t)}} \cdot \overbrace{\boxed{\frac{\partial h^{(t)}}{\partial h^{(t-1)}}}}^{W_h} \cdot \frac{\partial h^{(t-1)}}{\partial W_h} \cdots + \frac{\partial J^{(t)}(\theta)}{\partial \hat{y}^{(t)}} \cdot \frac{\partial \hat{y}^{(t)}}{\partial h^{(t)}} \cdot \overbrace{\boxed{\frac{\partial h^{(t)}}{\partial h^{(t-1)}} \cdot \cdots \cdot \frac{\partial h^{(2)}}{\partial h^{(1)}}}}^{(W_h)^{t-1}} \cdot \frac{\partial h^{(1)}}{\partial W_h}$$

동일한 가중치(W_h) 공유

- ➢ W_h가 작을 수록(< 1) 반복적으로 곱해지는 값이 0에 가까워져 gradient vanishing
- ➢ W_h가 클 수록(> 1) 반복적으로 곱해지는 값이 기하급수적으로 커져 gradient exploding

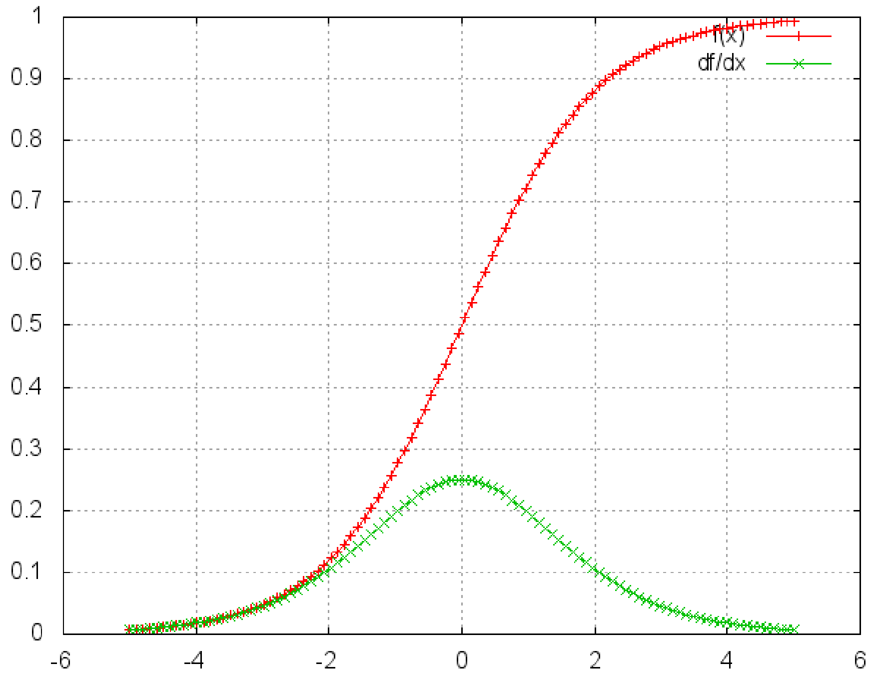
03 Problem of RNN

Sigmoid vs. Tanh

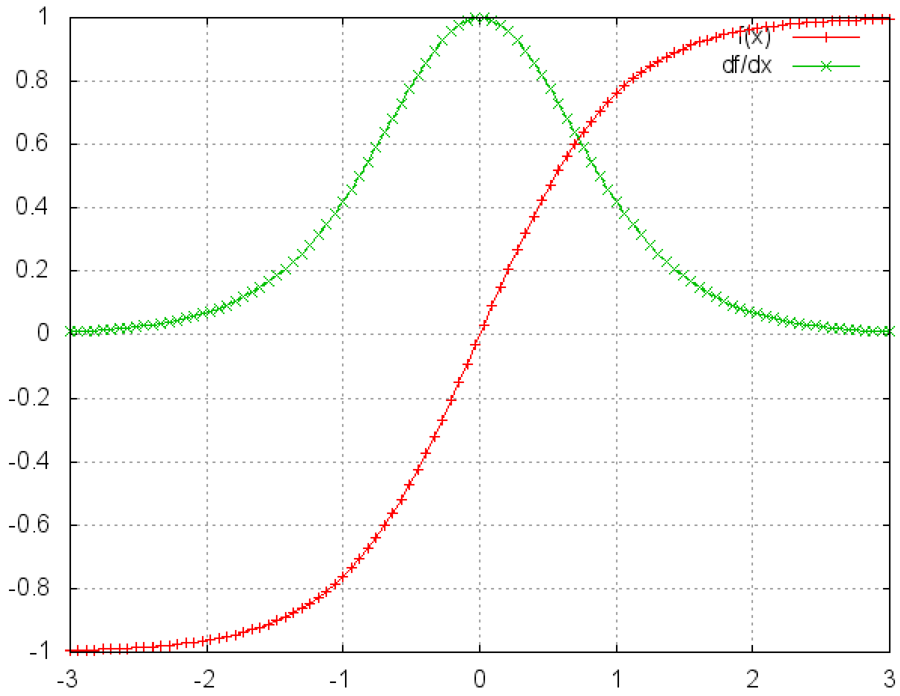
③ Hidden States

$$\boldsymbol{h}^{(t)} = \sigma\left(\boldsymbol{W}_h \boldsymbol{h}^{(t-1)} + \boldsymbol{W}_e \boldsymbol{e}^{(t)} + \boldsymbol{b}_1\right)$$

Sigmoid



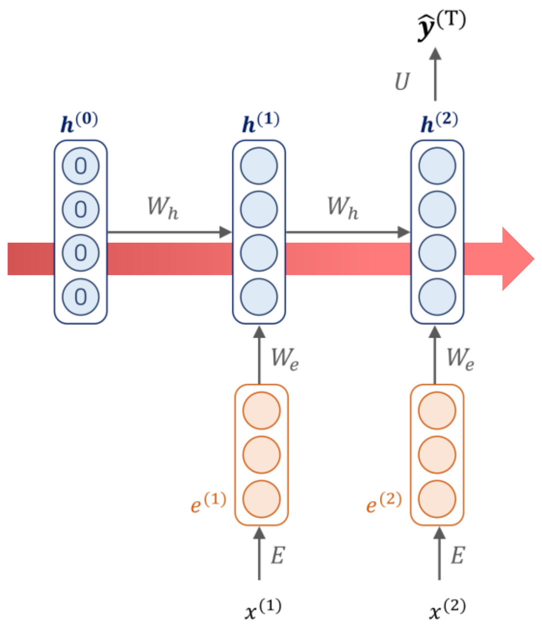
Tanh



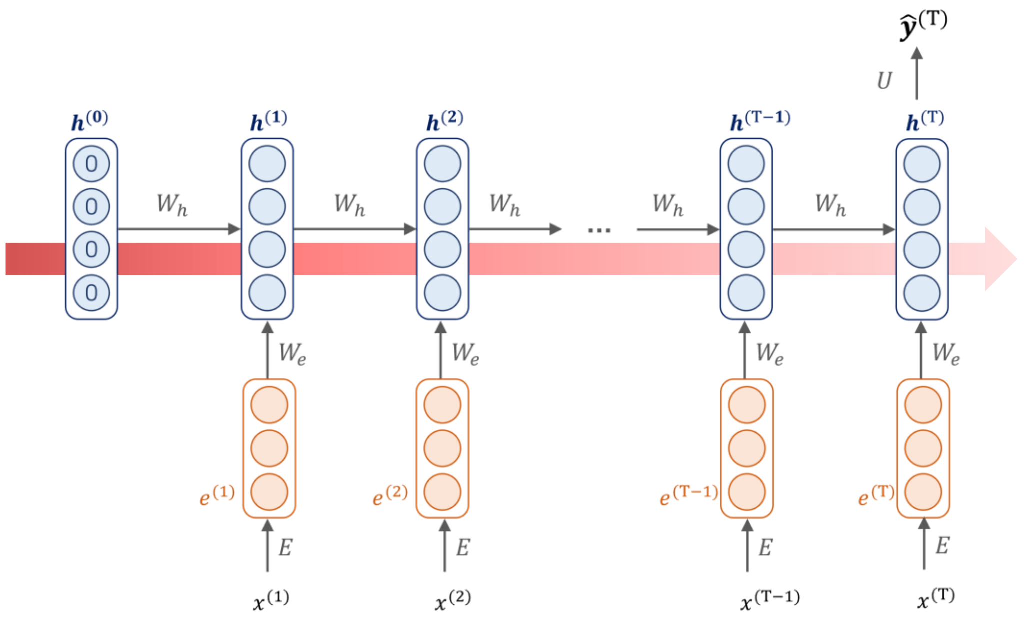
- ✓ **Sigmoid**: 기울기 0에서 약 0.25 사이
- ✓ **Tanh**: 기울기 0에서 1 사이 -> gradient vanishing에 더 강함

03 Problem of RNN

② Long Term Dependency (장기 의존성)



시퀀스 길이: 2

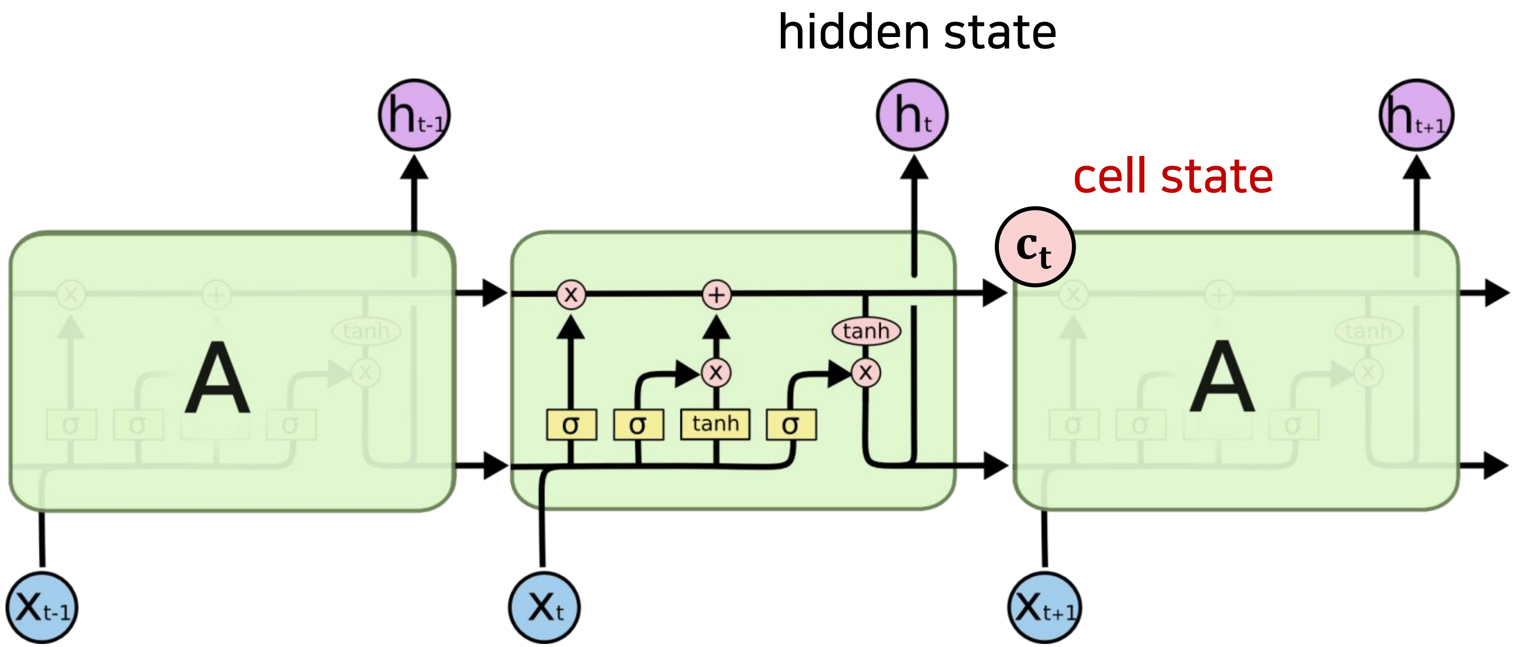


시퀀스 길이: T

- input sequence 길이가 긴 경우, 시퀀스 초반의 정보가 후반 time step의 hidden state를 도출하는데까지 전달되기 어려움 -> **Long term dependency를 잘 반영하지 못함**

04 LSTM: Long Short-Term Memory RNN

RNN with Separate Memory

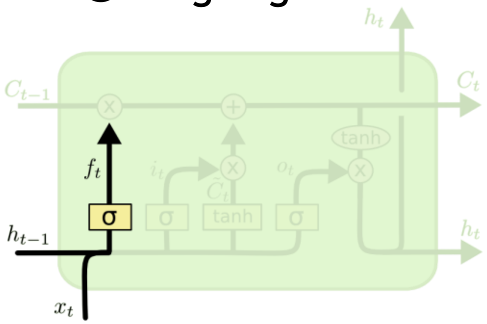


- **hidden state**를 통해 short term memory를 조절하고 **cell state**를 통해 long term memory 보존
- **forget, input, output, 3개의 gate**를 통해 매 time step의 cell state와 hidden state, input에서 취할 정보의 양 결정

LSTM: Long Short-Term Memory RNN

Architecture

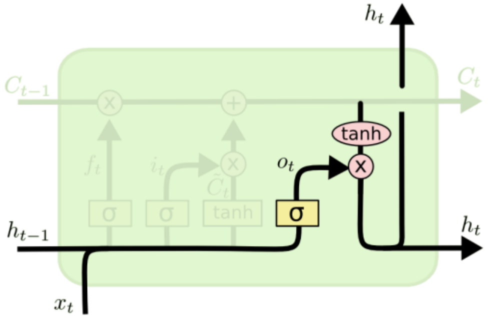
① forget gate



$$f_t = \sigma(W_f h^{(t-1)} + U_f x^{(t)} + b_f)$$

- 입력 정보(새로운 입력 시퀀스와 이전 시점의 hidden state)에 시그모이드를 취함

② input gate, new cell content



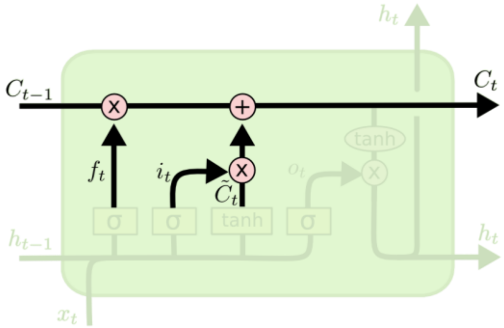
$$i_t = \sigma(W_i h^{(t-1)} + U_i x^{(t)} + b_i)$$

- input gate: 입력 정보에 시그모이드를 취함

$$\widetilde{C}_t = \tanh(W_c h^{(t-1)} + U_c x^{(t)} + b_c)$$

- new cell content: 입력 정보에 tanh를 취함

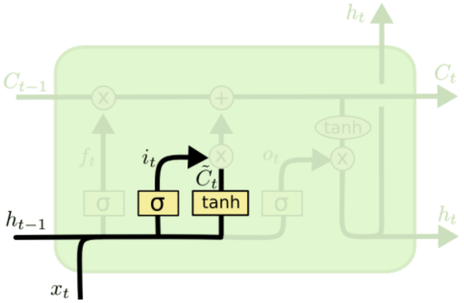
③ Cell state



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

- forget gate: 이전 시점의 cell state에서 어느 정도의 정보를 가져갈 것인지 결정
- input gate: 입력 정보에서 장기 기억으로 가져갈 정보의 양 결정
- element wise product

④ Output gate, hidden state



$$o_t = \sigma(W_o h^{(t-1)} + U_o x^{(t)} + b_o)$$

- output gate: 입력 정보에 시그모이드를 취함

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

- hidden state: 현재 입력과 대비해서 장기 기억에서 어느 정도의 정보를 단기 기억으로 사용할지 결정

05 Summary

Conclusion

✓ **Language Model** : 확률 분포를 기반으로 주어진 문맥(sequence) 이후에 위치할 단어 예측

- **N-gram LM**: 이전에 등장하는 N개의 단어 chunck를 바탕으로 다음에 올 단어 예측 (sparsity problem)
- **Neural LM**: 이전에 등장하는 window size만큼의 단어를 바탕으로 다음에 올 단어 예측
- input length 에 제한 -> 모든 문맥 고려 불가능

✓ **RNN**

- input length 에 제한이 없는 sequential neural network
- gradient vanishing /exploding
- long term dependency problem

✓ **LSTM**

- cell state를 통해 long term memory 보존
- 3개의 gate를 통해 정보의 flow 조절

감사합니다