

# 2022학년도 1학기 컴퓨터언어학

## 제15강 시퀀스 처리 (2)

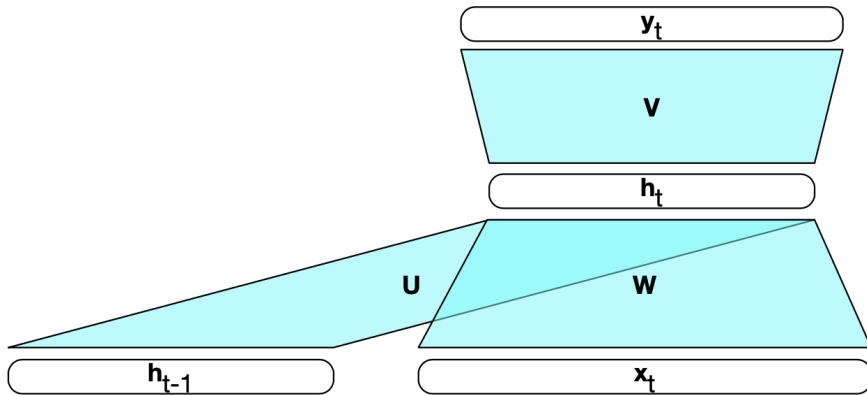
박수지

서울대학교 인문대학 언어학과

2022년 4월 27일 수요일

## 오늘의 목표

- 1 순방향 신경망을 순환 신경망으로 확장하는 과정을 설명할 수 있다.
- 2 순환 신경망을 자연언어처리에 활용하는 방법을 설명할 수 있다.



**Figure 9.3** Simple recurrent neural network illustrated as a feedforward network.

## 순방향 신경망

$$\vec{h} = g(\mathbf{W}\vec{x} + \vec{b})$$

$\vec{x}$  입력층

$\vec{h}$  은닉층

$g$  활성화 함수 (주로 ReLU)

## 순환 신경망

$$\vec{h}_t = g(\mathbf{U}\vec{h}_{t-1} + \mathbf{W}\vec{x}_t + \vec{b})$$

$t$  시간

$\vec{x}_t$  현재 시점  $t$ 에서의 입력층

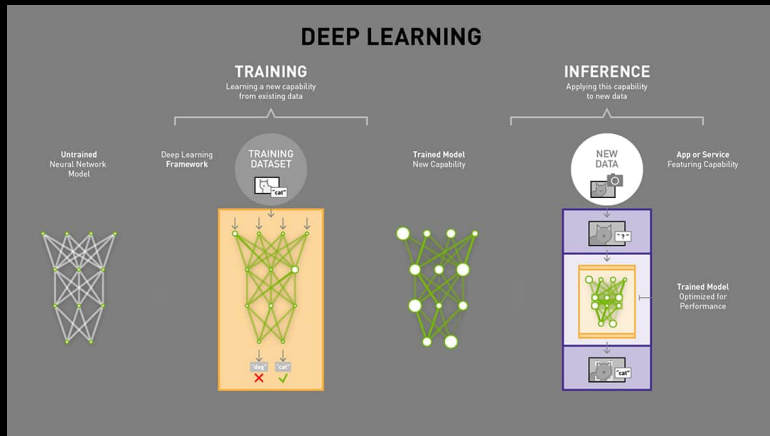
$\vec{h}_{t-1}$  직전 시점  $(t - 1)$ 에서의 은닉층

$\vec{h}_t$  현재 시점  $t$ 에서의 은닉층

$g$  활성화 함수 (주로 tanh)

## 딥러닝의 두 단계

- 1 훈련(Training)
- 2 추론(Inference)



<https://blogs.nvidia.com/blog/2016/08/22/difference-deep-learning-training-inference-ai/>

## 2층 단순 순환 신경망의 추론: 분류 문제의 경우

(현재 단계의) 입력  $\vec{x}_t \in \mathbb{R}^{d_{in}}$

직전 단계의 은닉층  $\vec{h}_{t-1} \in \mathbb{R}^{d_h}$

현재 단계의 은닉층  $\vec{h}_t = \tanh(\mathbf{U}\vec{h}_{t-1} + \mathbf{W}\vec{x}_t) \in \mathbb{R}^{d_h}$  (편향 생략)

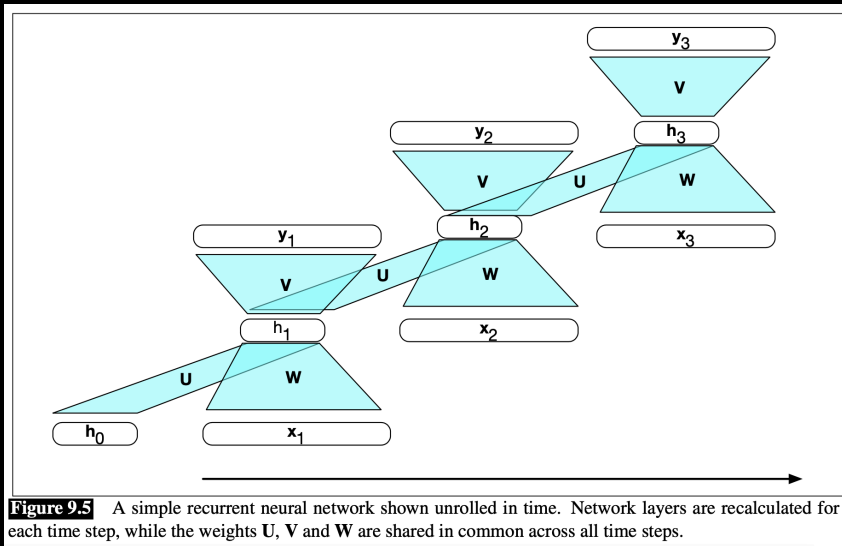
(현재 단계의) 출력  $\vec{y}_t = \text{softmax}(\mathbf{V}\vec{h}_t) \in \mathbb{R}^{d_{out}}$

## 가중치 매개변수 행렬의 차원

입력층  $\rightarrow$  은닉층  $\mathbf{W} \in \mathbb{R}^{d_h \times d_{in}}$

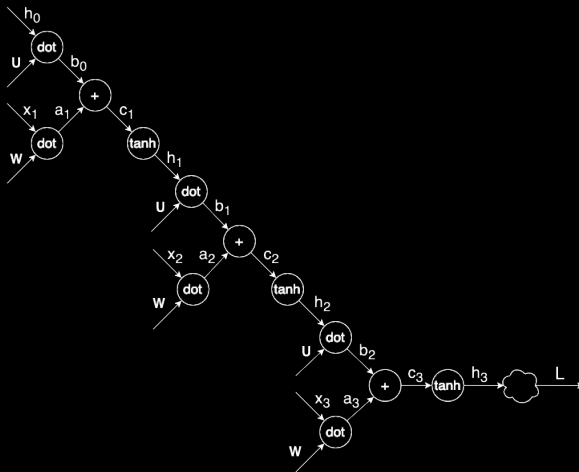
은닉층  $\rightarrow$  은닉층  $\mathbf{U} \in \mathbb{R}^{d_h \times d_h}$

은닉층  $\rightarrow$  출력층  $\mathbf{V} \in \mathbb{R}^{d_{out} \times d_h}$



# 순환 신경망의 계산 그래프

입력의 길이가 3인 경우





## 순환 신경망 언어 모형

**입력 시퀀스**  $\mathbf{X} = [\vec{x}_1; \cdots; \vec{x}_t; \cdots \vec{x}_N]$  (원-핫 인코딩)

**임베딩 계층**  $\vec{e}_t = \mathbf{E}\vec{x}_t$

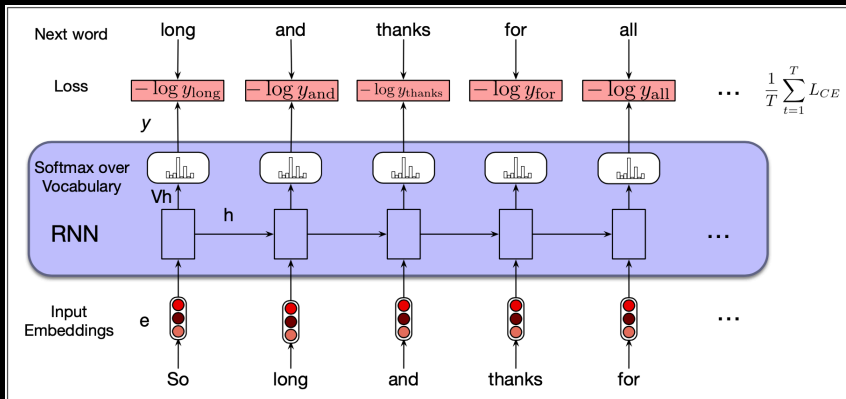
**은닉층**  $\vec{h}_t = \tanh(\mathbf{U}\vec{h}_{t-1} + \mathbf{W}\vec{e}_t)$

**출력층**  $\vec{y}_t = \text{softmax}(\mathbf{V}\vec{h}_t) \in \mathbb{R}^{|V|}$

$P(w_{t+1} = i | w_1, \cdots, w_t) = y_t[i]$ 로 표기하면

$$\begin{aligned} P(w_{1:n}) &= \prod_{i=1}^n P(w_i | w_{1:(i-1)}) \quad (\text{연쇄 법칙}) \\ &= \prod_{i=1}^n y_i[w_i] \quad (\text{위에서 정의한 표기법}) \end{aligned}$$

$$L_{CE} = - \sum_{i \in V} y_t[i] \log \hat{y}_t[i] = - \log \hat{y}_t[w_{t+1}]$$



**Figure 9.6** Training RNNs as language models.

## 순환 신경망 언어 모형

입력 시퀀스  $\mathbf{X} = [\vec{x}_1; \cdots; \vec{x}_t; \cdots \vec{x}_N]$  (원-핫 인코딩)

임베딩 계층  $\vec{e}_t = \mathbf{E}\vec{x}_t$

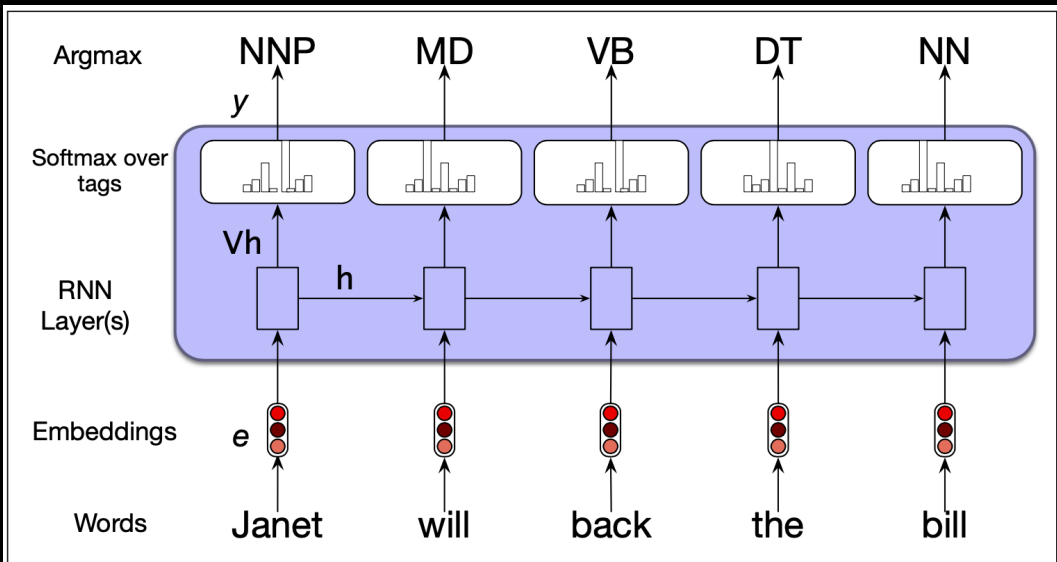
은닉층  $\vec{h}_t = \tanh(\mathbf{U}\vec{h}_{t-1} + \mathbf{W}\vec{e}_t)$

출력층  $\vec{y}_t = \text{softmax}(\mathbf{V}\vec{h}_t) \in \mathbb{R}^{|V|}$

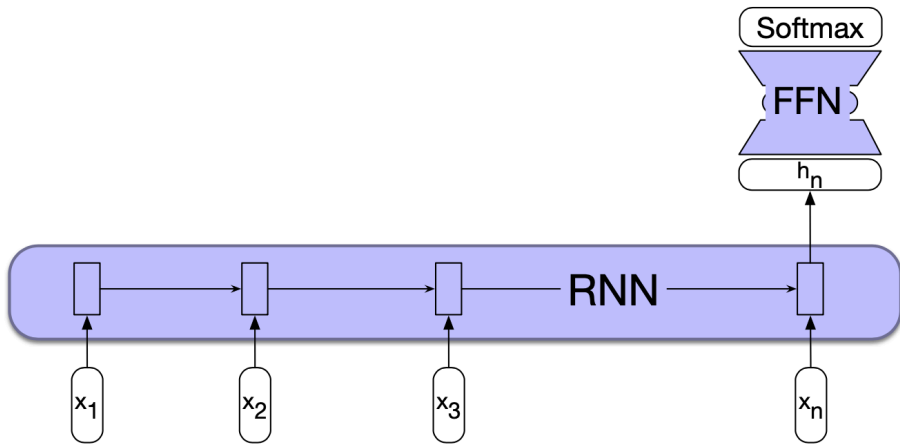
의문: 출력층에서  $\mathbf{V}$ 를 따로 둘 필요가 있는가?

## 순환 신경망 언어 모형 (수정)

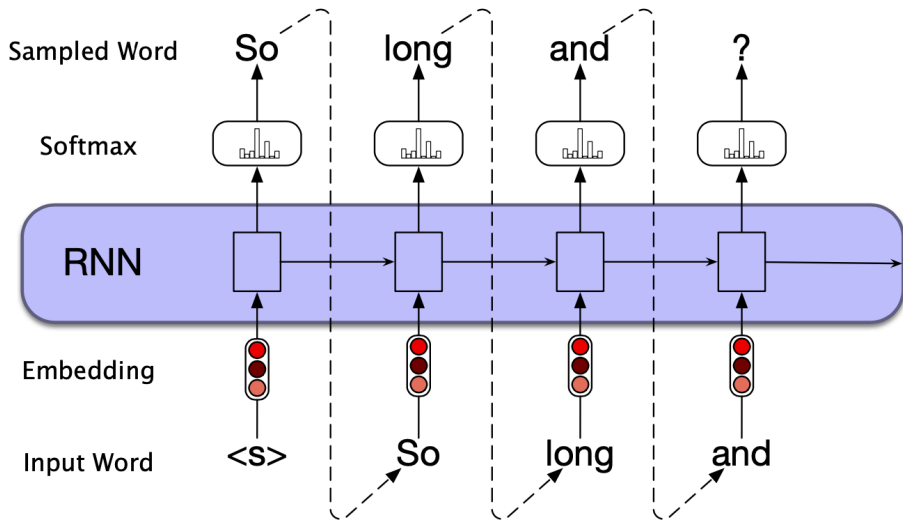
출력층  $\vec{y}_t = \text{softmax}(\mathbf{E}^\top \vec{h}_t) \in \mathbb{R}^{|V|}$



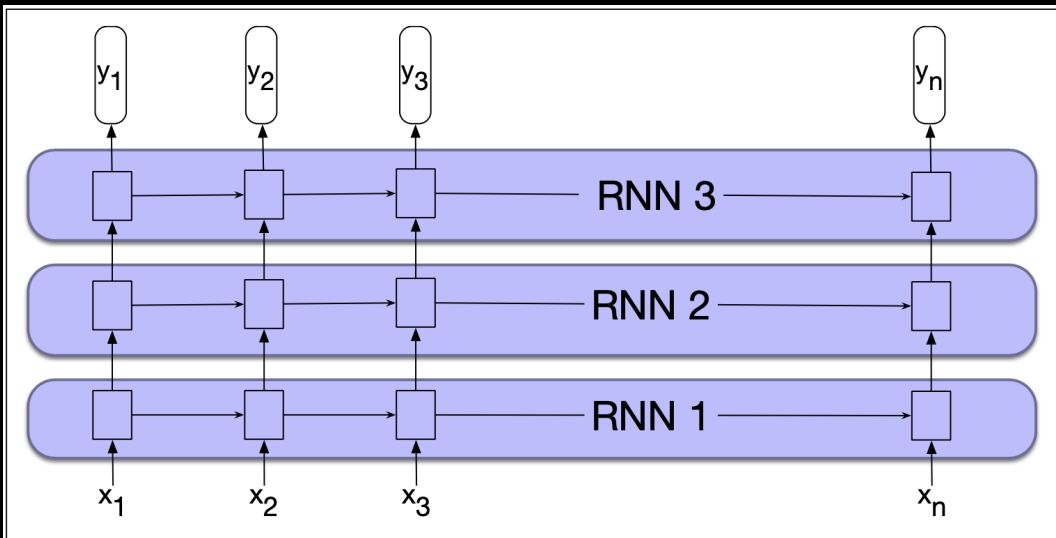
**Figure 9.7** Part-of-speech tagging as sequence labeling with a simple RNN. Pre-trained



**Figure 9.8** Sequence classification using a simple RNN combined with a feedforward network. The final hidden state from the RNN is used as the input to a feedforward network that performs the classification.



**Figure 9.9** Autoregressive generation with an RNN-based neural language model.



**Figure 9.10** Stacked recurrent networks. The output of a lower level serves as the input to higher levels with the output of the last network serving as the final output.

남은 문제

순환 신경망 훈련의 어려움: 기울기 소실 문제

해결책

LSTM(Long-short term memory)