

5장. 딥러닝 – IV

4. 재귀 신경망

4.1 재귀 신경망

4.2 ReLU 활성화 함수를 사용하는 재귀 신경망

4.3 LSTM 재귀 신경망

4.4 GRU 재귀 신경망

4.5 재귀 신경망의 확장

4.6 재귀 신경망의 적용분야

4.1 재귀 신경망

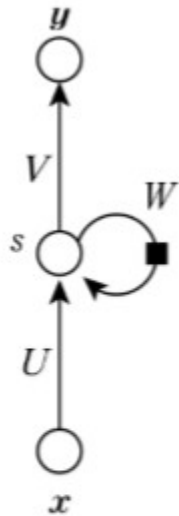
❖ 재귀 신경망(Recurrent Neural Networks, RNN, 순환 신경망)

- **서열 데이터**(Sequence data)
 - 음성, 자연어 문장, 동영상, 주가 변동 등의 데이터
 - 구성요소가 순차적으로 발생하거나 구성요소 간에 순서 존재
 - 이전 값들이 현재 값에 영향을 주는 경우
- 서열 데이터의 **분류, 예측**에서 **현재 시점의 값과 이전 시점의 값들**을 고려 필요
- 재귀 신경망은 서열 데이터의 학습 및 추론에 적합한 모델
- 기계 번역, 음성 인식, 필기체 인식, 영상 주석달기, 동영상에서 행동 인식, 작곡 및 작사 등 다양한 응용 분야에서 활용

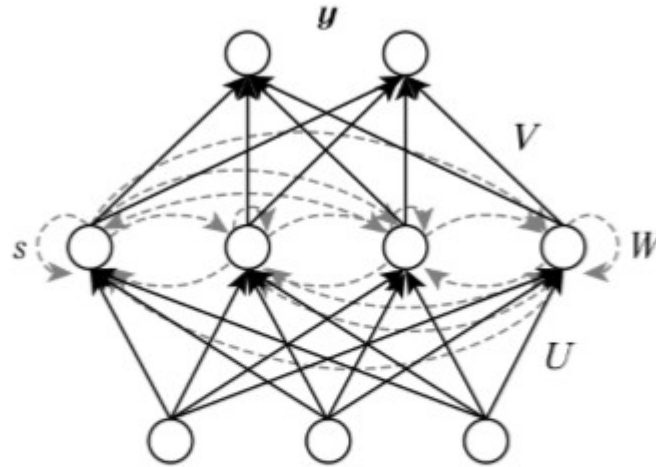
4.1.1 재귀 신경망의 구조와 동작

❖ 재귀 신경망의 구조

- 기본적으로 **은닉층** 한 개와 **출력층**으로 구성 입력의 일부로
- 과거의 정보를 반영하기 위해, **은닉층 또는 출력층의 값을** 입력의 일부로 **사용**



(a)



(b)

그림 5.52 RNN 모델 (a) RNN 모델의 형태 (b) RNN 모델의 실제 형태

재귀 신경망의 구조와 동작

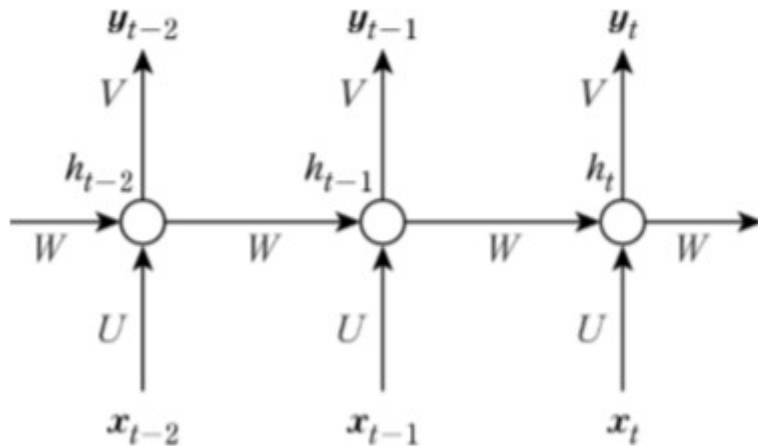
❖ 재귀 신경망의 동작

$$s_t = Ux_t + Wh_{t-1} + b_s$$

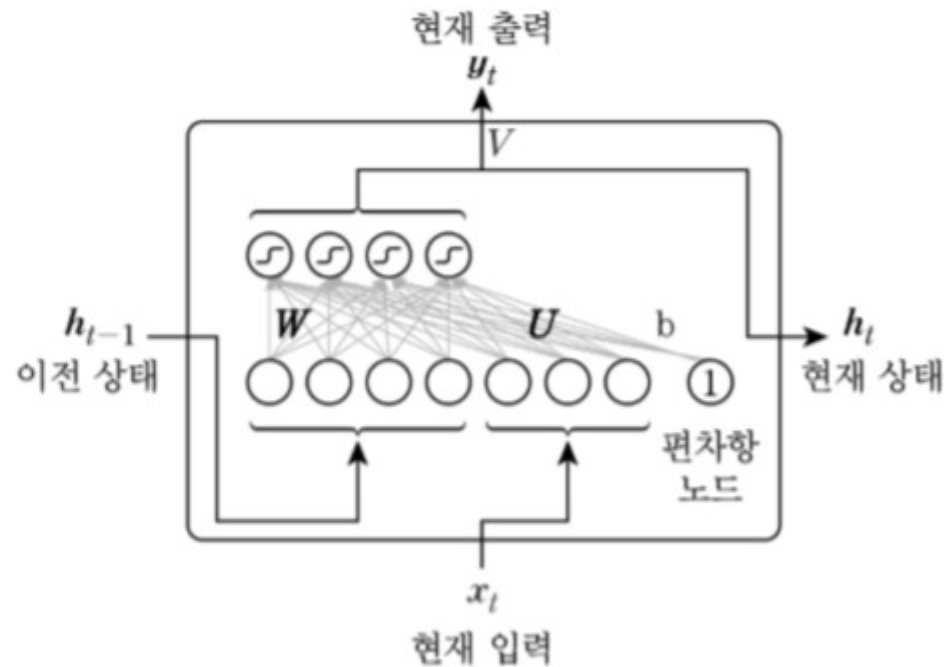
$$h_t = f(s_t)$$

$$z_t = Vh_t + b_z$$

$$y_t = g(z_t)$$



(a)

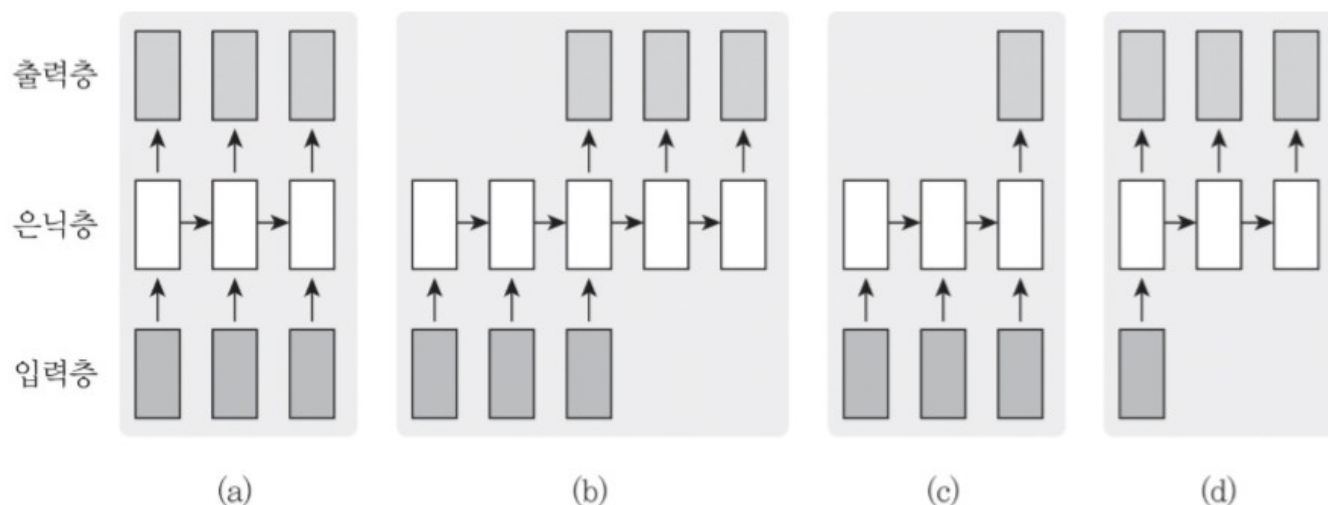


(b)

그림 5.53 RNN 모델 (a) RNN을 펼쳐놓은 형태 (b) RNN의 입출력 구조

재귀 신경망의 구조와 동작

재귀 신경망에서 입력과 출력의 대응 형태



(a) 각 시점의 입력에 대한 출력이 학습 데이터에 지정

(b) 앞 시점에 입력이 끝나면서 출력값이 주어지는 상황

- 기계 번역: '이것은 책이다' → 'This is a book'

(c) 일련의 데이터가 입력으로 주어진 다음, 마지막에 결과 값이 주어지는 상황

- 감성 분석: '이 책은 내용이 알차게 구성되어 있다' → '긍정적'

(d) 하나의 입력에 대해 일련의 출력 이 나오는 것

- 영상 주석달기: 영상 → 설명하는 문장

4.1.2 재귀 신경망의 학습

❖ 재귀 신경망의 학습 데이터 형태

- 서열 데이터의 집합
- Ex. 문자열 'hello'의 학습 데이터 형태
 - $h \rightarrow e, e \rightarrow l, l \rightarrow l, l \rightarrow o$

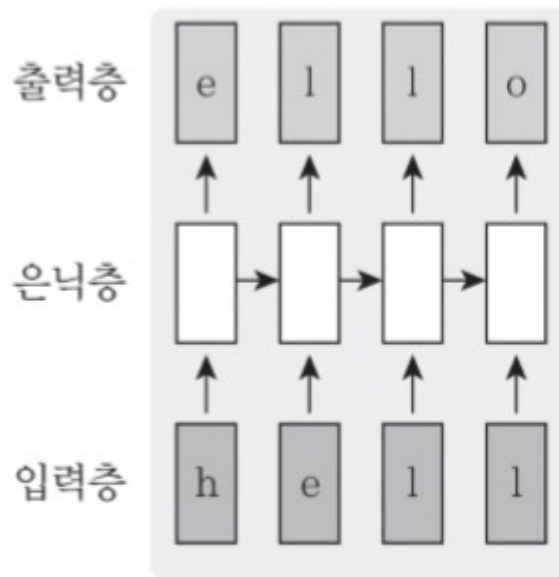


그림 5.55 RNN으로 'hello'를 학습할 때 학습 데이터

재귀 신경망의 학습

❖ BPTT(Back Propagation Through Time) 알고리즘

- 과거 시간으로 오차를 전달하여 가중치 조정
- 과거 시점으로 오차를 전달할 때 각 가중치는 동일하게 사용
- 학습을 할 때는, 각 시점에서의 그래디언트를 구한 다음, 그 평균값을 해당 변수에 대한 그래디언트로 사용

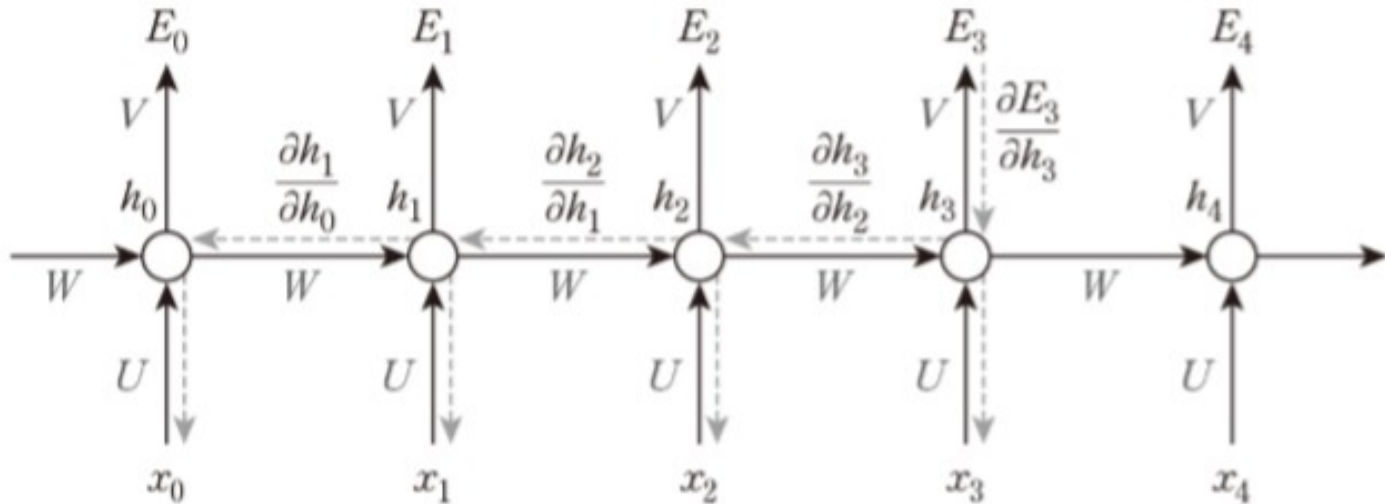


그림 5.56 $t = 3$ 에서 각 이전 시점으로 역전파되는 오차 정보의 흐름

재귀 신경망의 학습

❖ BPTT 알고리즘 – Cont.

- 목표 출력 서열
- RNN의 출력 서열

$$\mathbf{y}' = (y'_0, y'_1, y'_2, \dots, y'_{N'})$$

$$\mathbf{y}_t = (y_{t1}, y_{t2}, \dots, y_{tK})$$

- 오차 함수

- 분류

$$E_t(\mathbf{y}_t, \mathbf{y}'_t) = - \sum_{i=1}^K y'_{ti} \log y_{ti}$$

- 회귀

$$E_t(\mathbf{y}_t, \mathbf{y}'_t) = \frac{1}{K} \sum_{k=1}^K (y'_{tk} - y_{tk})^2$$

- 목적함수

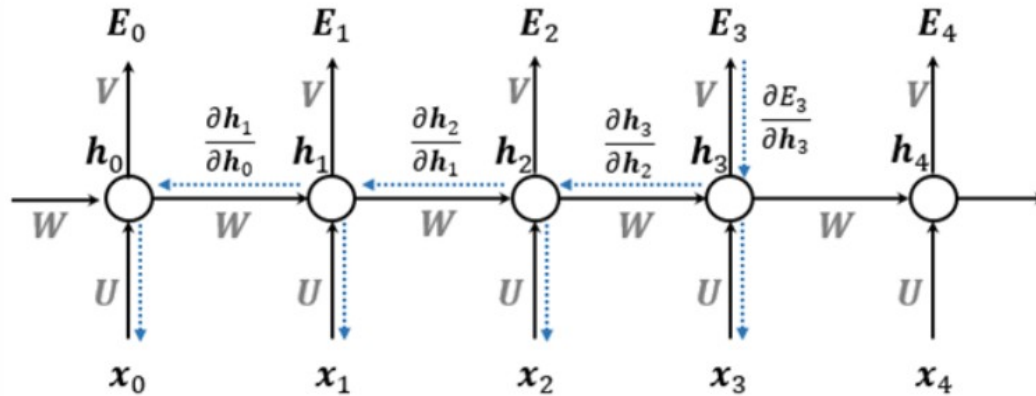
$$E(\mathbf{y}, \mathbf{y}') = \sum_t E_t(\mathbf{y}_t, \mathbf{y}'_t)$$

- 목적 함수의 그레디언트

$$\frac{\partial E}{\partial W} = \sum_i \frac{E_t(\mathbf{y}_t, \mathbf{y}'_t)}{\partial W}$$

재귀 신경망의 학습

✧ BPTT 알고리즘 – Cont.



- $t = 3$ 일 때 W 에 대한 그레디언트

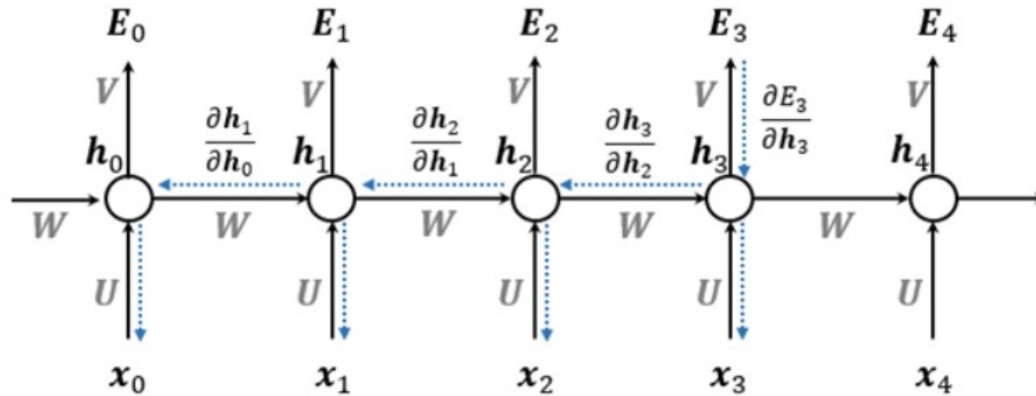
$$\frac{\partial E_3}{\partial W} = \frac{\partial E_3}{\partial y_3} \frac{\partial y_3}{\partial h_3} \frac{\partial h_3}{\partial W}$$

$$\frac{\partial E_3}{\partial W} = \sum_{k=0}^3 \frac{\partial E_3}{\partial y_3} \frac{\partial y_3}{\partial h_3} \frac{\partial h_3}{\partial h_k} \frac{\partial h_k}{\partial W}$$

$$= \frac{\partial E_3}{\partial y_3} \frac{\partial y_3}{\partial h_3} \left[\frac{\partial h_3}{\partial h_3} \frac{\partial h_3}{\partial W} + \frac{\partial h_3}{\partial h_2} \frac{\partial h_2}{\partial W} + \frac{\partial h_3}{\partial h_1} \frac{\partial h_1}{\partial W} + \frac{\partial h_3}{\partial h_0} \frac{\partial h_0}{\partial W} \right]$$

재귀 신경망의 학습

✧ BPTT 알고리즘 – Cont.



- $t = 3$ 일 때 V 에 대한 그래디언트

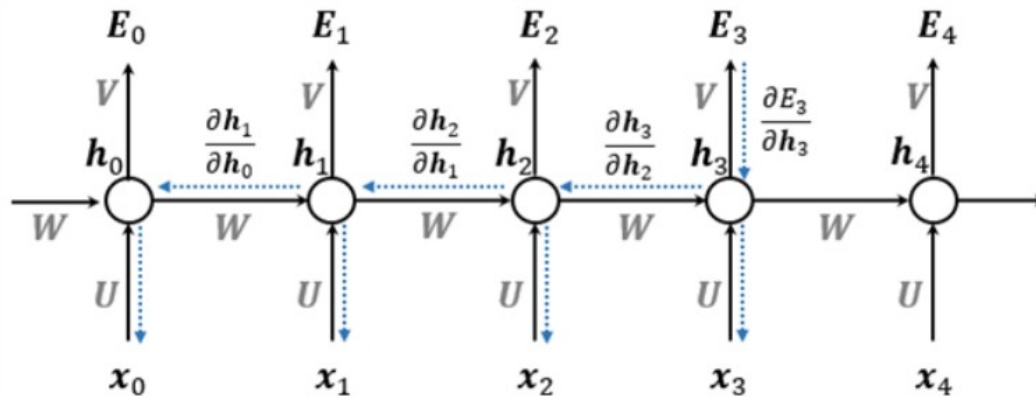
$$\frac{\partial E_3}{\partial V} = \frac{\partial E_3}{\partial y_3} \frac{\partial y_3}{\partial V} = \frac{\partial E_3}{\partial y_3} \frac{\partial y_3}{\partial z_3} \frac{\partial z_3}{\partial V}$$

- $t = 3$ 일 때 U 에 대한 그래디언트

$$\frac{\partial E_3}{\partial U} = \sum_{k=0}^3 \frac{\partial E_3}{\partial y_3} \frac{\partial y_3}{\partial h_3} \frac{\partial h_3}{\partial h_k} \frac{\partial h_k}{\partial U}$$

4.1.3 재귀 신경망의 기울기 소멸과 폭발

❖ 그래디언트



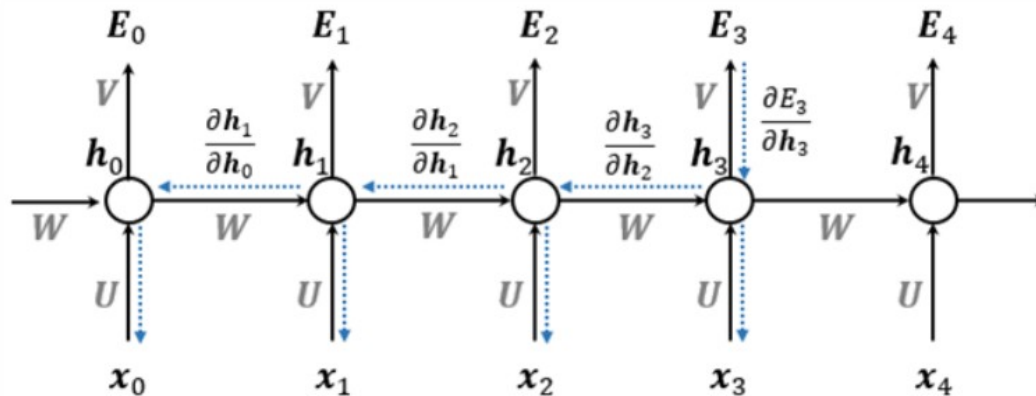
$$\frac{\partial E_3}{\partial W} = \sum_{k=0}^3 \frac{\partial E_3}{\partial \mathbf{y}_3} \frac{\partial \mathbf{y}_3}{\partial \mathbf{h}_3} \frac{\partial \mathbf{h}_3}{\partial \mathbf{h}_k} \frac{\partial \mathbf{h}_k}{\partial W} = \sum_{k=0}^3 \frac{\partial E_3}{\partial \mathbf{y}_3} \frac{\partial \mathbf{y}_3}{\partial \mathbf{h}_3} \left(\prod_{j=k+1}^3 \frac{\partial \mathbf{h}_j}{\partial \mathbf{h}_{j-1}} \right) \frac{\partial \mathbf{h}_k}{\partial W}$$

$$\mathbf{h}_j = f(\mathbf{s}_j) = f(U\mathbf{x}_j + W\mathbf{h}_{j-1} + \mathbf{b}_s)$$

$$\frac{\partial \mathbf{h}_j}{\partial \mathbf{h}_{j-1}} = W^\top \text{diag}(f'(\mathbf{s}_j))$$

재귀 신경망의 기울기 소멸과 폭발

✧ 그레디언트 – cont.



$$\prod_{j=1}^3 \frac{\partial h_j}{\partial h_{j-1}} = \underline{W^\top} \text{diag}(f'(s_3)) \underline{W^\top} \text{diag}(f'(s_2)) \underline{W^\top} \text{diag}(f'(s_1))$$

- $t = 100$ 이면, W^\top 의 100 거듭제곱 포함
- $f(x) = x$ 라고 가정할 때
 - $f'(x) = 1 \rightarrow \text{diag}(f'(s)) = I$

$$\prod_{j=1}^{100} \frac{\partial h_j}{\partial h_{j-1}} = (W^\top)^{100}$$

재귀 신경망의 기울기 소멸과 폭발

✧ 그레디언트 – cont.

- 행렬 W 의 고유값 분해

$$W = Q\Lambda Q^{-1} \quad \Lambda : \text{고유값을 대각 원소로 갖는 대각 행렬}$$
$$Q : \text{고유벡터를 열벡터로 갖는 행렬}$$

$$(W^T)^{100} = (Q^{-1})^T \Lambda^{100} Q^T$$

- ex. $\Lambda = \begin{pmatrix} -0.6 & 0 \\ 0 & 1.8 \end{pmatrix} \quad \Lambda^3 = \begin{pmatrix} (-0.6)^3 & 0 \\ 0 & 1.8^3 \end{pmatrix} = \begin{pmatrix} -0.216 & 0 \\ 0 & 3.375 \end{pmatrix}$
$$\Lambda^{10} = \begin{pmatrix} 0.006 & 0 \\ 0 & 357,046 \end{pmatrix}$$

- $|\text{고유값}| < 1$ 이면, 기울기 소멸
- $|\text{고유값}| > 1$ 이면, 기울기 폭발

재귀 신경망의 기울기 소멸과 폭발

❖ 기울기 소멸 문제(Vanishing gradient problem)

- 오차 정보를 역전파 시키는 과정에서 그래디언트가 급격히 영벡터에 가까워져서 학습이 되지 않는 현상
- 일부 가중치 성분에서만 발생 가능 □ 문제 발생 파악 곤란

❖ 기울기 폭발 문제(Exploding gradient problem)

- 학습과정에 그래디언트가 급격히 커지는 현상
- 일부 성분에서의 기울기 폭발현상은 다른 성분에도 바로 파급
□ 문제 발생 확인 용이

4.1.4 재귀 신경망의 기울기 소멸과 폭발 조건

✧ 재귀 신경망의 기울기 소멸과 폭발 조건

- λ_1 : 가중치 행렬 W 의 최대 고유값
- $\lambda_1 < 1/\gamma$ 일 때
 - 기울기 소멸 문제 반드시 발생
 - 활성화 함수가 $\tanh()$ 일 때, $\gamma = 1$
 - 활성화 함수가 $\sigma()$ 일 때, $\gamma = 1/4$
- $\lambda_1 > 1/\gamma$ 일 때
 - 기울기 폭발 문제 발생 가능

4.1.5 기울기 폭발 문제의 대응 방법

✧ 기울기 폭발 문제의 대응 방법

▪ RMSprop 방법 사용

- 최근 그레디언트들의 크기의 평균에 해당 하는 값으로 나누어 사용
- 그레디언트의 갑작스러운 큰 변화를 방지하는 효과
- 기울기 소멸 문제뿐만 아니라 기울기 폭발 문제 완화

▪ 단기 BPTT(truncated BPTT) 사용

- 오차정보를 최근 몇 단계까지만 역전파
- 가중치 행렬 W 이 거듭제곱되는 횟수 제한

▪ 그레디언트 최대값 고정 방법 사용

- 그레디언트가 일정한 임계값 이상이 되면 임계값 으로 고정
- $\|\nabla f\| > \theta$ 이면

$$\nabla f \leftarrow \theta \frac{\nabla f}{\|\nabla f\|}$$

4.2 LeRU 활성화 함수를 사용하는 재귀 신경망

❖ LeRU를 사용하는 재귀 신경망

▪ IRNN

- 은닉층에서 은닉층으로의 가중치를 나타내는 행렬 W 를 **항등행렬** / 로 초기화한 후에 학습
- 기존의 RNN 모델보다 높은 성능 개선 가능

▪ np-RNN

- 은닉층에서 은닉층으로의 가중치를 고유값의 하나는 1이고 나머지는 1보다 작은 값은 갖는 **양의 준정부호 행렬**(positive semi-definite matrix)로 초기화

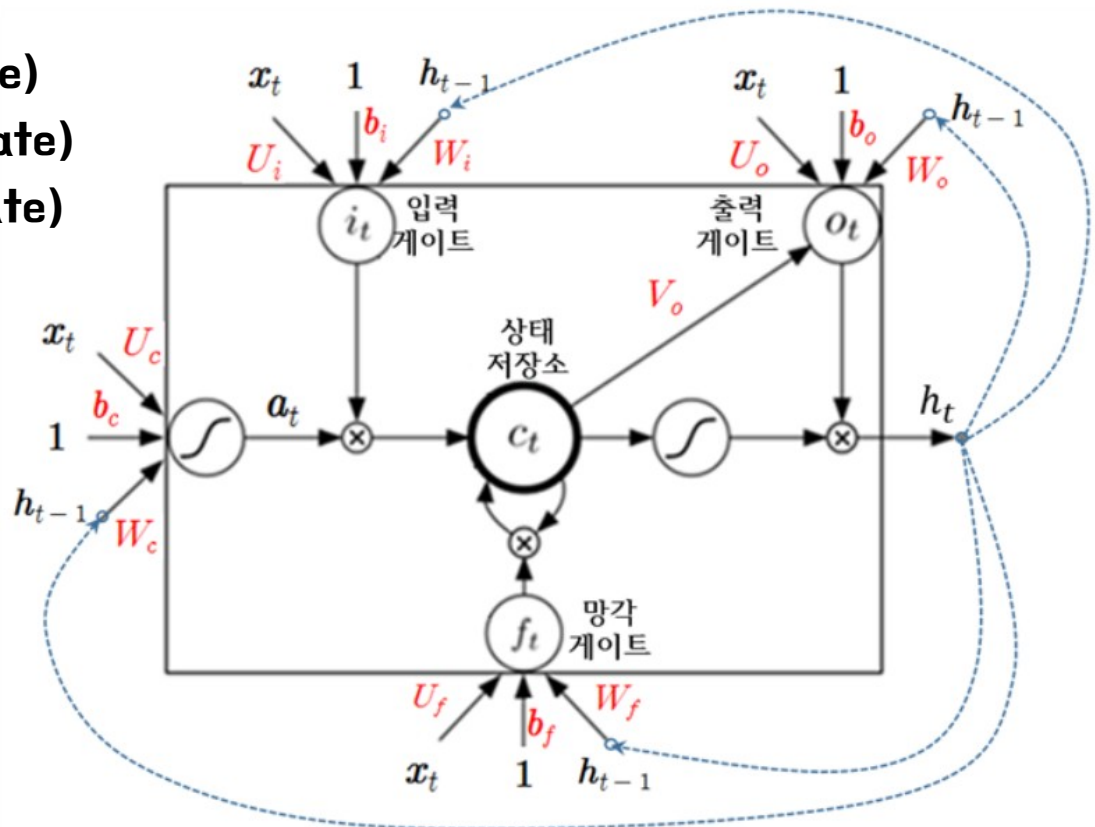
▪ uRNN

- 은닉층의 가중치 W 가 모든 고유값의 크기가 1인 **유니타리 행렬** (unitary matrix)이 되도록 하면서 학습

4.3 LSTM 재귀 신경망

❖ LSTM(Long Short Time Memory) RNN

- 역전파되는 그래디언트가 쉽게 소멸되는 현상을 완화시키는 RNN 모델
- 각 은닉 노드가 상태 저장소(memory cell)와 저장, 출력, 망각을 조절하는 게이트(gate) 포함
 - 입력 게이트(input gate)
 - 출력 게이트(output gate)
 - 망각 게이트(forget gate)



4.3.1 LSTM 재귀 신경망의 구조

❖ 일반 재귀 신경망의 입력에 대한 민감도

- 시점 $t = 1$ 에서의 입력에 대한 시점별 민감도를 노드의 진하기로 보인 것
- 시간이 진행됨에 따라 새로운 입력이 은닉 상태에 반영되기 때문에 과거의 기억은 점차 사라짐

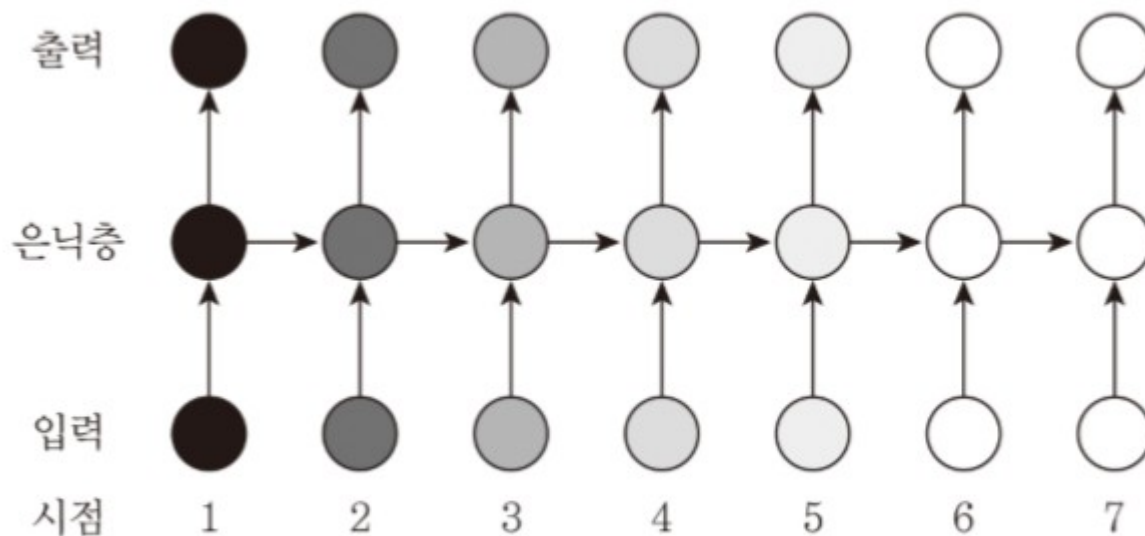


그림 5.57 재귀 신경망에서 시점 1의 입력에 대한 그레디언트의 민감도

LSTM 재귀 신경망의 구조

❖ 게이트 장착을 통한 민감도 조절

- 게이트의 동작을 통해 먼 시점까지 영향 전파 가능
- LSTM은 게이트의 동작을 학습을 통해 결정

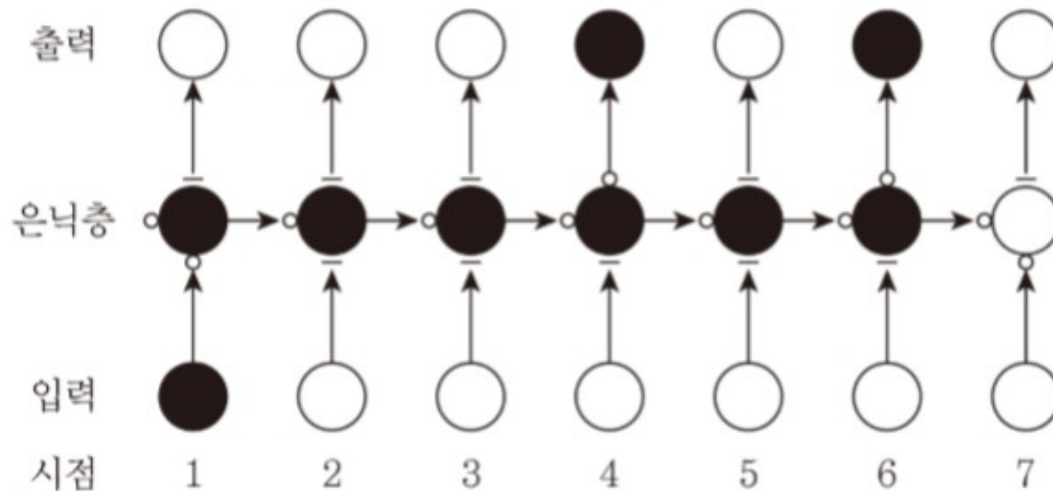
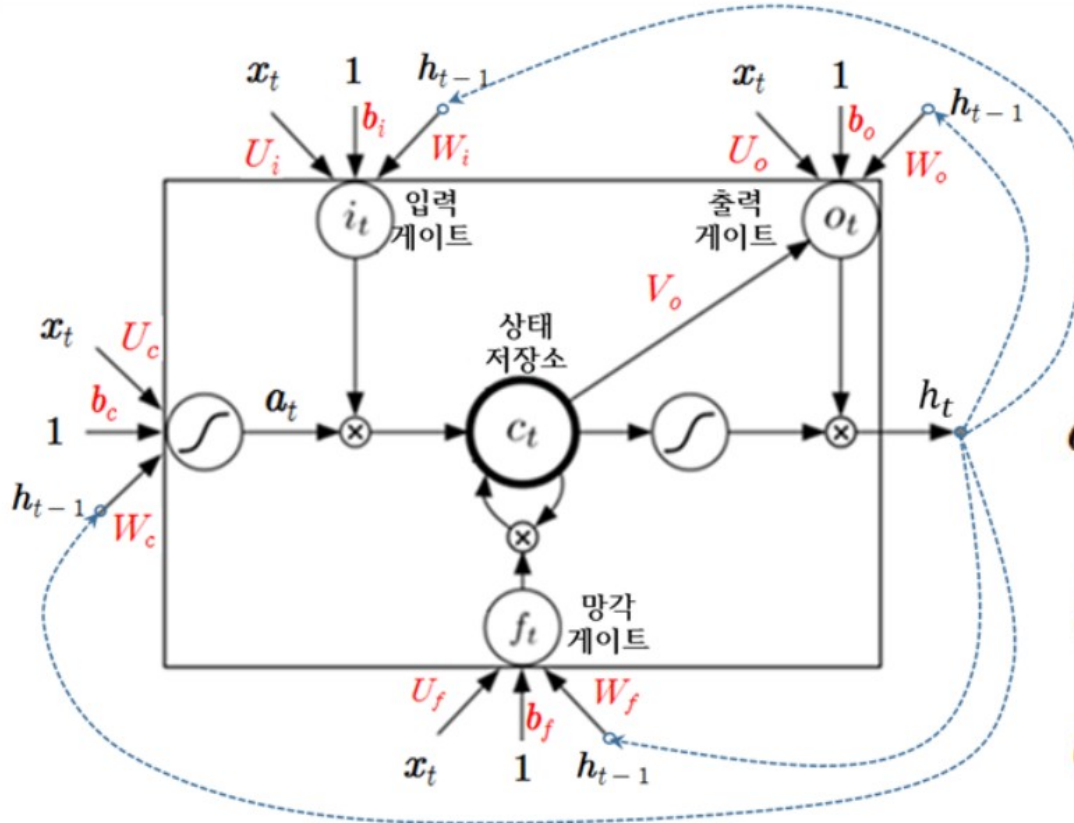


그림 5.58 게이트의 열림-닫힘에 의한 시점 1의 입력에 대한 그레디언트의 민감도

노드의 왼쪽에는 망각 게이트, 아래쪽에는 입력 게이트, 위쪽에는 출력 게이트가 위치하고, O는 게이트의 열림, - 또는 |는 게이트의 닫힘을 나타낸다.

4.3.2 LSTM 재귀 신경망의 동작

❖ LSTM 재귀 신경망의 동작



$$i_t = \sigma(U_i x_t + W_i h_{t-1} + b_i)$$

$$a_t = \tanh(U_c x_t + W_c h_{t-1} + b_c)$$

$$f_t = \sigma(U_f x_t + W_f h_{t-1} + b_f)$$

$$c_t = i_t \circ a_t + f_t \circ c_{t-1}$$

$$o_t = \sigma(U_o x_t + W_o h_{t-1} + V_o c_{t-1} + b_o)$$

$$h_t = o_t \circ \tanh(c_t)$$

LSTM 재귀 신경망의 동작

알고리즘 5.3 LSTM 재귀 신경망의 실행

입력 : 서열 데이터 (x_1, x_2, \dots, x_T)

가중치 $W_c, U_c, W_i, U_i, W_f, U_f, W_o, U_o, V_o$

편차항 b_c, b_i, b_f, b_o

출력 : LSTM의 출력 (h_1, h_2, \dots, h_T)

1. $h_0 \leftarrow 0$
 2. $c_0 \leftarrow 0$
 3. for $t = 1$ to T
 4. $i_t \leftarrow \sigma(U_i x_t + W_i h_{t-1} + b_i)$
 5. $a_t \leftarrow \tanh(U_c x_t + W_c h_{t-1} + b_c)$
 6. $f_t \leftarrow \sigma(U_f x_t + W_f h_{t-1} + b_f)$
 7. $c_t \leftarrow i_t \circ a_t + f_t \circ c_{t-1}$
 8. $o_t \leftarrow \sigma(U_o x_t + W_o h_{t-1} + V_o c_{t-1} + b_o)$
 9. $h_t \leftarrow o_t \circ \tanh(c_t)$
-

4.3.3 LSTM 재귀 신경망의 학습

✧ LSTM 재귀 신경망의 학습

- BPTT 사용
- 게이트와 상태 저장소에 대한 오차 함수 E 의 그레디언트

$$\boxed{\frac{\partial E}{\partial c_t^k}} = \frac{\partial E}{\partial h_t^k} \frac{\partial h_t^k}{\partial c_t^k} = \frac{\partial E}{\partial h_t^k} o_t^k (1 - \tanh^2(c_t^k))$$

$$\frac{\partial E}{\partial i_t^k} = \boxed{\frac{\partial E}{\partial c_t^k}} \frac{\partial c_t^k}{\partial i_t^k} = \frac{\partial E}{\partial c_t^k} a_t^k$$

$$\frac{\partial E}{\partial f_t^k} = \boxed{\frac{\partial E}{\partial c_t^k}} \frac{\partial c_t^k}{\partial f_t^k} = \frac{\partial E}{\partial c_t^k} c_{t-1}^k$$

$$\frac{\partial E}{\partial a_t^k} = \boxed{\frac{\partial E}{\partial c_t^k}} \frac{\partial c_t^k}{\partial a_t^k} = \frac{\partial E}{\partial c_t^k} i_t^k$$

$$\frac{\partial E}{\partial o_t^k} = \frac{\partial E}{\partial h_t^k} \frac{\partial h_t^k}{\partial o_t^k} = \frac{\partial E}{\partial h_t^k} \tanh(c_t^k)$$

LSTM 재귀 신경망의 학습

✧ LSTM 재귀 신경망의 학습 – cont.

- 시점 $t - 1$ 로 역전파되는 그래디언트

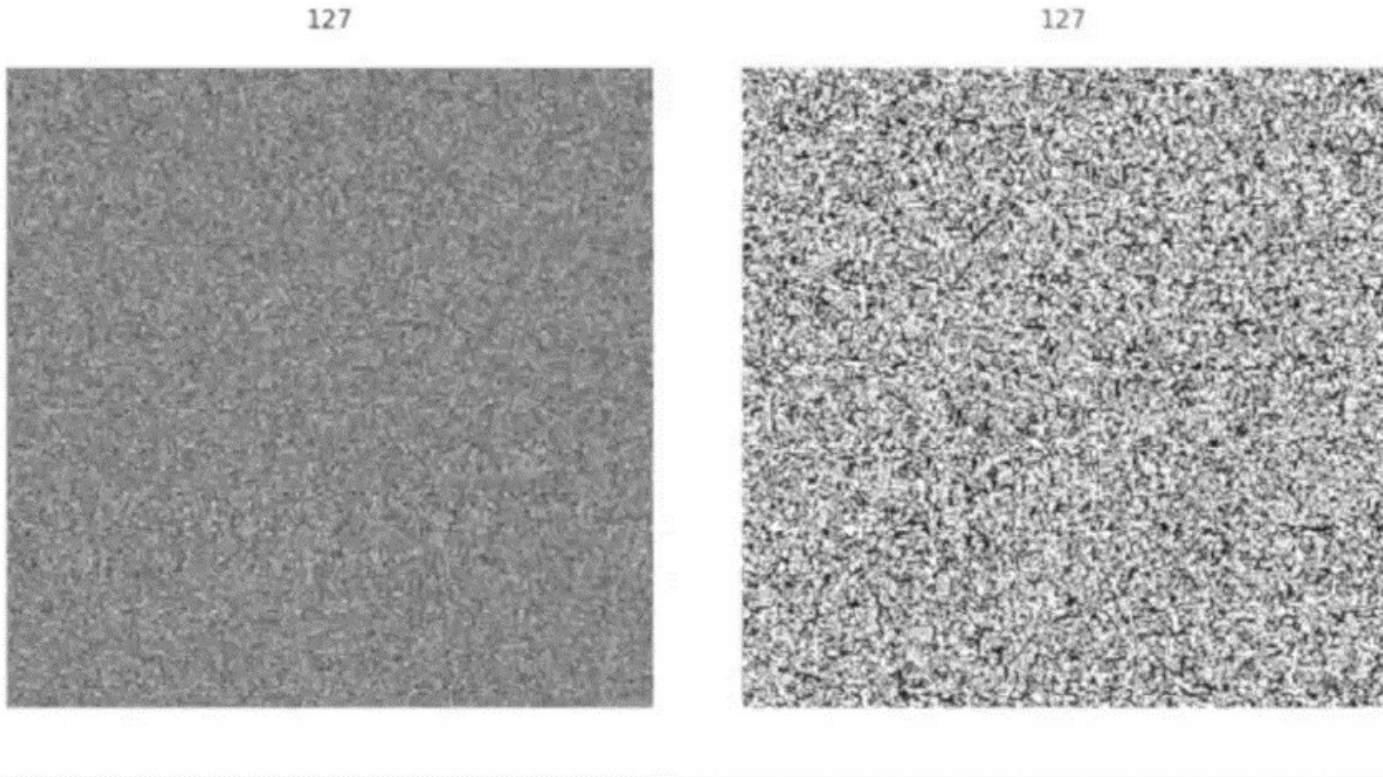
$$\frac{\partial E}{\partial c_{t-1}^k} = \frac{\partial E}{\partial c_t^k} \frac{\partial c_t^k}{\partial c_{t-1}^k} = \frac{\partial E}{\partial c_t^k} \frac{\partial (i_t^k a_t^k + f_t^k c_{t-1}^k)}{\partial c_{t-1}^k} = \frac{\partial E}{\partial c_t^k} f_t^k$$

- p 단계 과거로 전달되는 그래디언트

$$\frac{\partial E}{\partial c_{t-p}^k} = \frac{\partial E}{\partial c_t^k} \prod_{n=t-p}^t f_n^k$$

LSTM 재귀 신경망의 학습

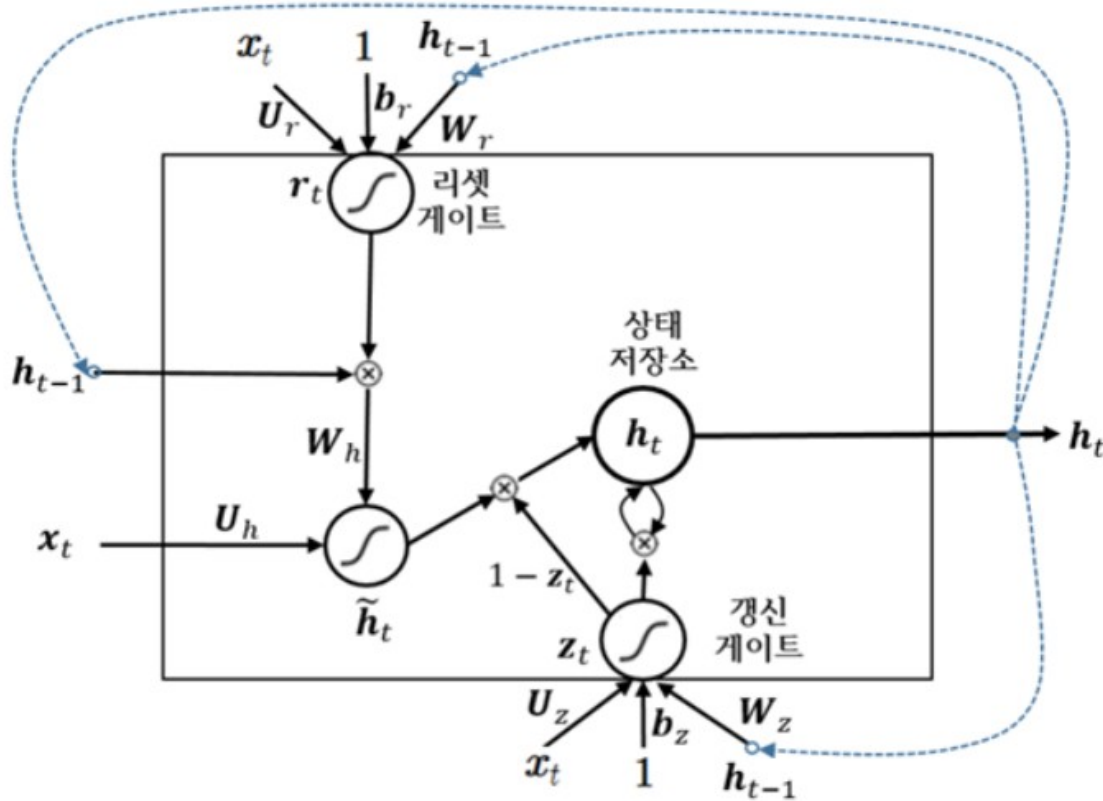
- ❖ 기존 재귀 신경망과 LSTM 재귀 신경망
 - 시간에 따른 그래디언트 전달 비교



4.4 GRU 재귀 신경망

❖ GRU(gated recurrent unit) 재귀 신경망

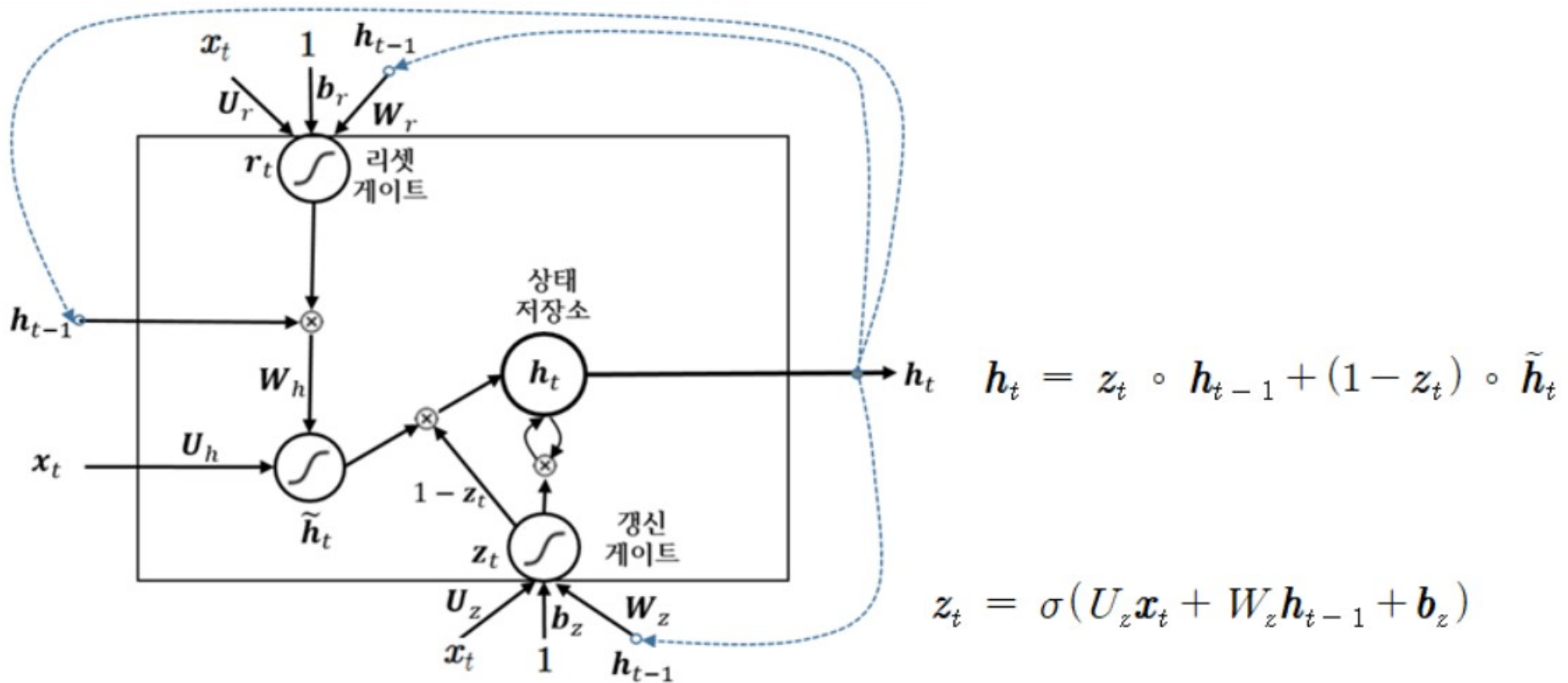
- 상태 저장소(memory cell)과 리셋 게이트(reset gate) 및 갱신 게이트(update gate) 포함 모델



GRU 재귀 신경망

❖ GRU 재귀 신경망의 동작

$$r_t = \sigma(U_r x_t + W_r h_{t-1} + b_r)$$



$$\tilde{h}_t = \tanh(U_h x_t + W_h (r_t \odot h_{t-1}))$$

GRU 재귀 신경망

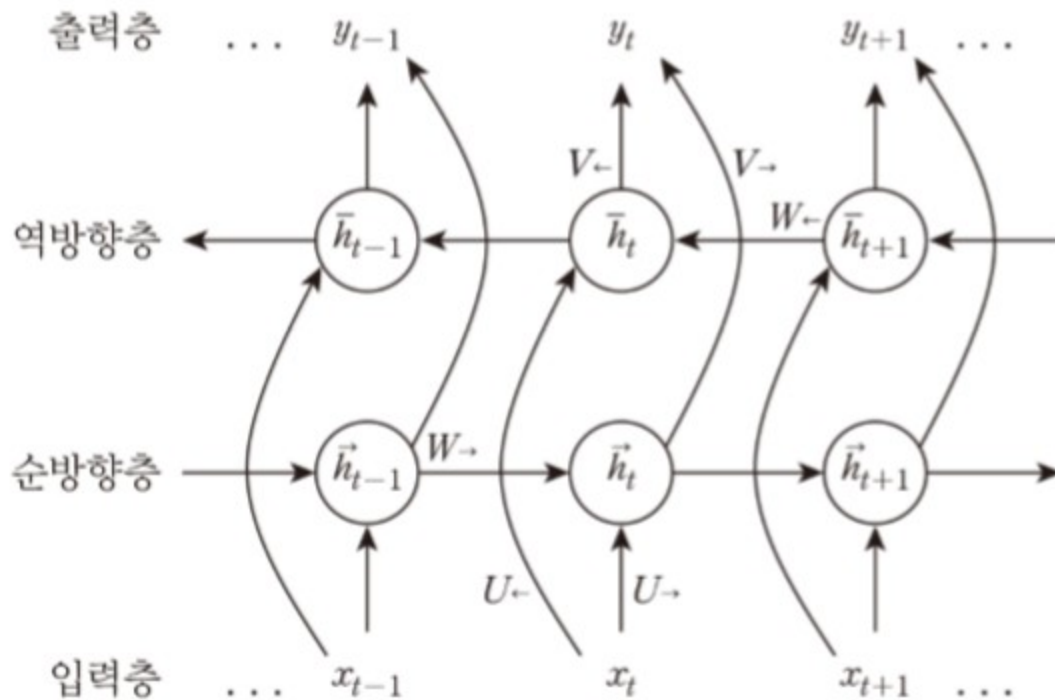
❖ GRU 재귀 신경망의 학습

- BPTT 알고리즘 사용
- LSTM 재귀 신경망의 파라미터 개수의 약 $\frac{3}{4}$ 파라미터 포함
- LSTM 재귀 신경망과 유사한 성능

4.5 재귀 신경망의 확장

❖ 양방향 재귀 신경망(Bidirectional RNN)

- 시점 t 의 출력이 이전 시점의 입력값과 은닉층의 값들 뿐만 아니라 이후 시점의 입력값과 은닉층의 값들에도 영향을 받도록 한 모델



$$y_t = f(V_{\rightarrow} \vec{h}_t + V_{\leftarrow} \bar{h}_t + b_o)$$

$$\bar{h}_t = \sigma(U_{\leftarrow} x_t + W_{\leftarrow} \bar{h}_{t+1} + b_{\leftarrow})$$

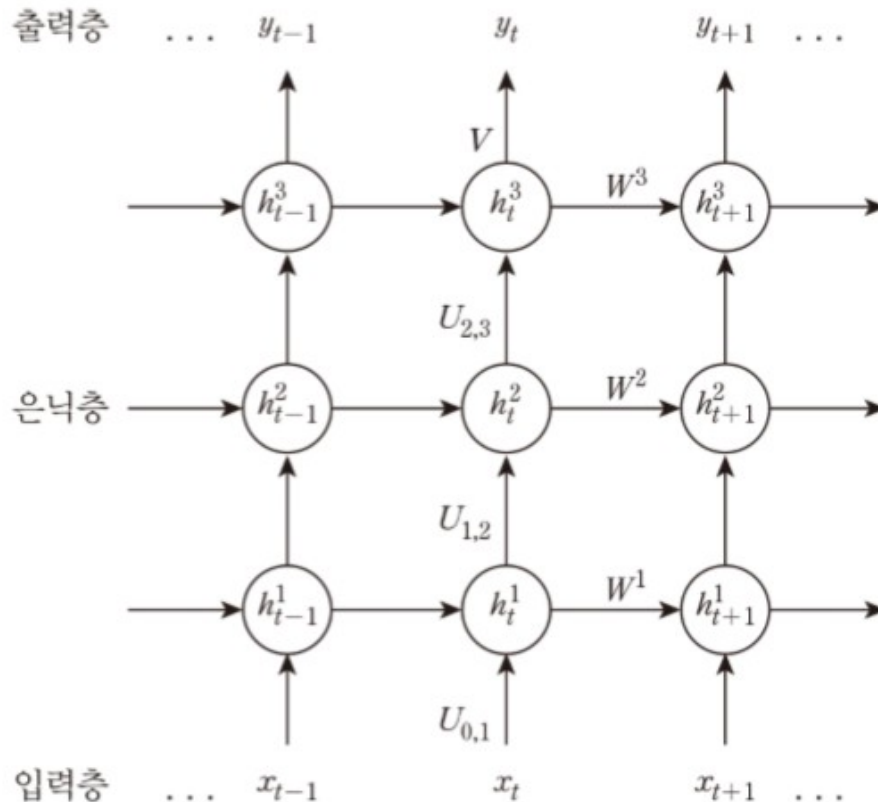
$$\vec{h}_t = \sigma(U_{\rightarrow} x_t + W_{\rightarrow} \vec{h}_{t-1} + b_{\rightarrow})$$

그림 5.61 양방향 재귀 신경망

재귀 신경망의 확장

❖ 딥러닝 재귀 신경망(Deep RNN)

- 여러 개의 재귀 신경망을 쌓아서 아래층의 출력을 바로 위층의 입력으로 받아들이도록 만든 모델



for $l = 1$ to L

$$h_0^l \leftarrow 0$$

for $t = 1$ to T

$$h_t^1 \leftarrow \sigma(U_{0,1}x_t + W^1h_{t-1}^1 + b_1)$$

for $l = 2$ to L

$$h_t^l \leftarrow \sigma(U_{l-1,l}h_{t-1}^{l-1} + W^lh_{t-1}^l + b_l)$$

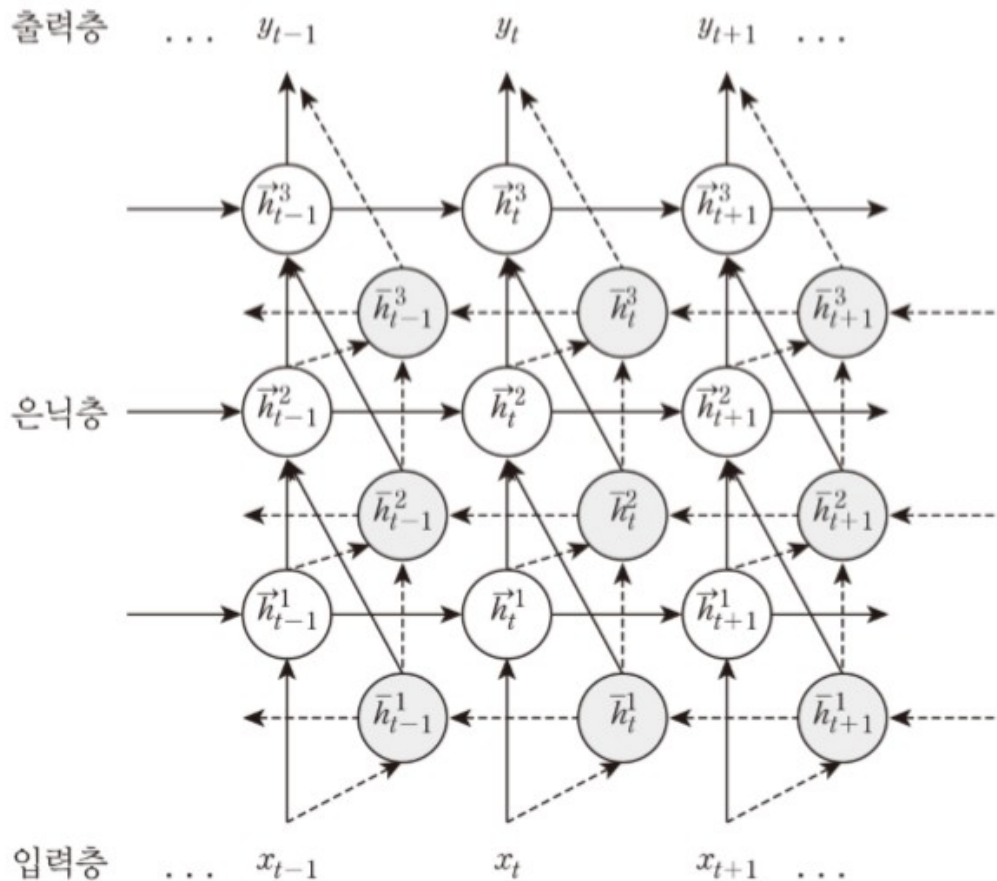
$$y_t \leftarrow f(Vh_t^L + b_o)$$

그림 5.62 딥러닝 재귀 신경망

재귀 신경망의 확장

❖ 딥러닝 양방향 재귀 신경망(Deep bidirectional RNN)

- 양방향 재귀 신경망과 딥러닝 재귀 신경망을 결합한 형태



$$y_t = g(V_{\rightarrow} \vec{h}_t^L + V_{\leftarrow} \bar{h}_t^L + b_o)$$

$$\vec{h}_t^l = f(\vec{U}_{\rightarrow}^l \vec{h}_t^{l-1} + \vec{U}_{\leftarrow}^l \bar{h}_t^{l-1} + W_{\rightarrow}^l \vec{h}_{t-1}^l + b_{\rightarrow}^l)$$

$$\bar{h}_t^l = f(\vec{U}_{\rightarrow}^l \vec{h}_t^{l-1} + \vec{U}_{\leftarrow}^l \bar{h}_t^{l-1} + W_{\leftarrow}^l \bar{h}_{t+1}^l + b_{\leftarrow}^l)$$

$$\vec{h}_t^1 = f(U_{\rightarrow}^1 x_t + W_{\rightarrow}^1 \vec{h}_{t-1}^1 + b_{\rightarrow}^1)$$

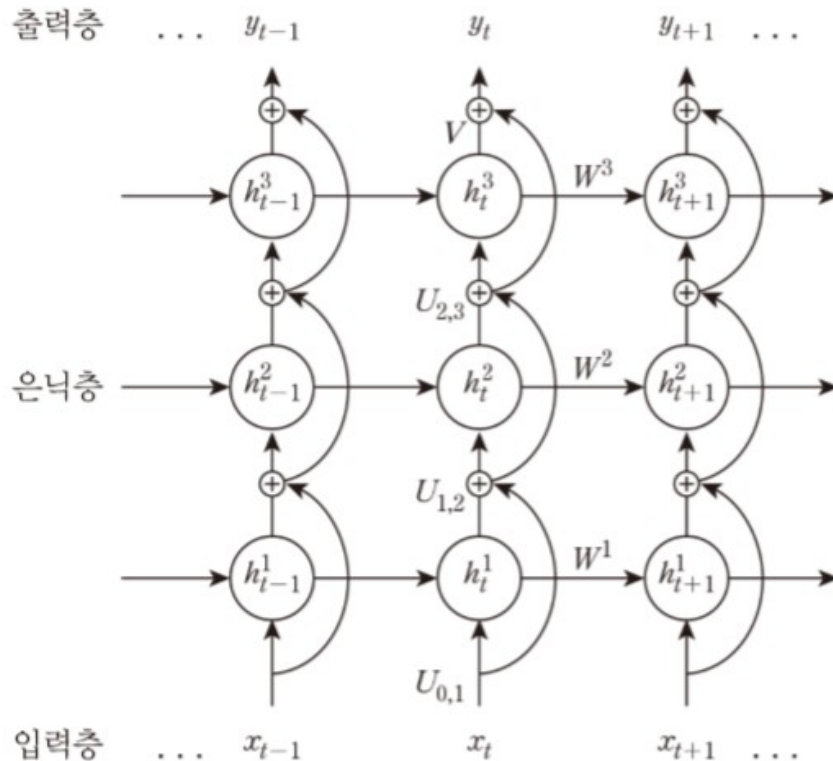
$$\bar{h}_t^1 = f(U_{\leftarrow}^1 x_t + W_{\leftarrow}^1 \bar{h}_{t+1}^1 + b_{\leftarrow}^1)$$

그림 5.63 딥러닝 양방향 재귀 신경망

재귀 신경망의 확장

❖ 잔차연결 딥러닝 재귀 신경망(Deep RNN with Residual Connection)

- 딥러닝 재귀 신경망에 층을 건너 뛸 수 있는 지름길 연결(skip connection)을 추가한 모델



$$h_t^1 \leftarrow \sigma(U_{0,1}x_t + W^1h_{t-1}^1 + b_1) + x_t$$

$$h_t^l \leftarrow \sigma(U_{l-1,l}h_{t-1}^{l-1} + W^lh_{t-1}^l + b_l) + h_{t-1}^l$$

그림 5.64 잔차연결 딥러닝 재귀 신경망

4.6 재귀 신경망의 적용 분야

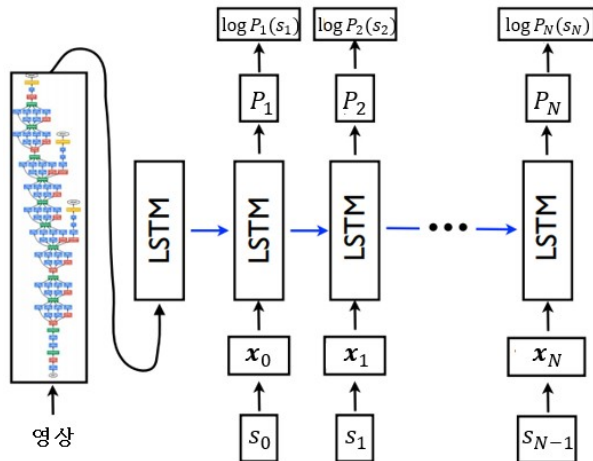
❖ 적용분야

■ 자연어 처리

- 언어 모델 (language model)
- 문장 생성
- 문서 생성
- 기계 번역 (machine translation)

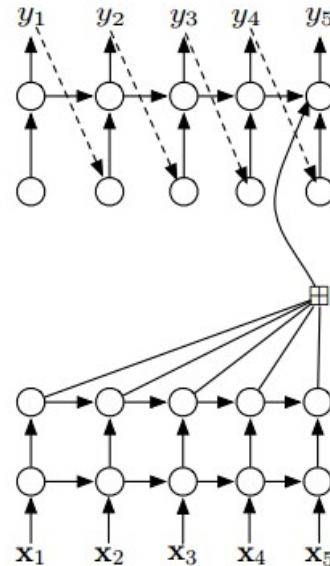
■ 음성 인식

■ 영상 주석달기 (Image captioning)

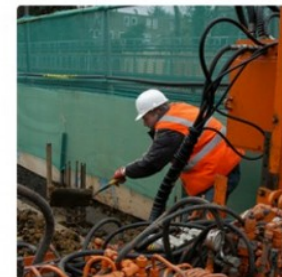


PANDARUS:

Alas, I think he shall be come approached and the day
When little strain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.



"man in black shirt is playing guitar."



"construction worker in orange safety vest is working on road."