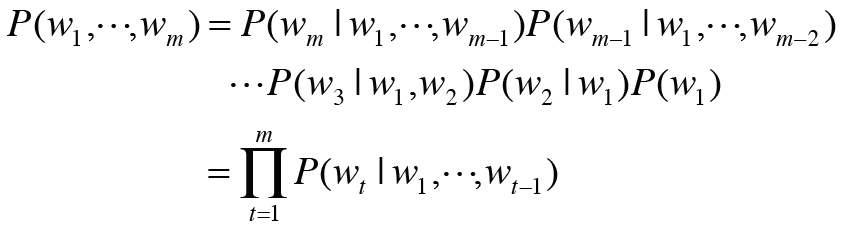
chapter 5 순환 신경망 (recurrent neural network: RNN)

시계열 데이터 처리 : 언어, 음성, 주가

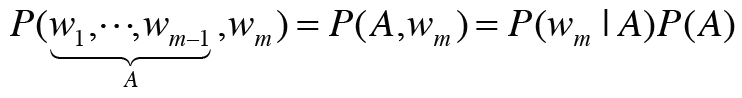
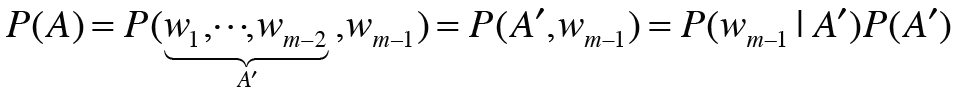
5.1 확률과 언어 모델

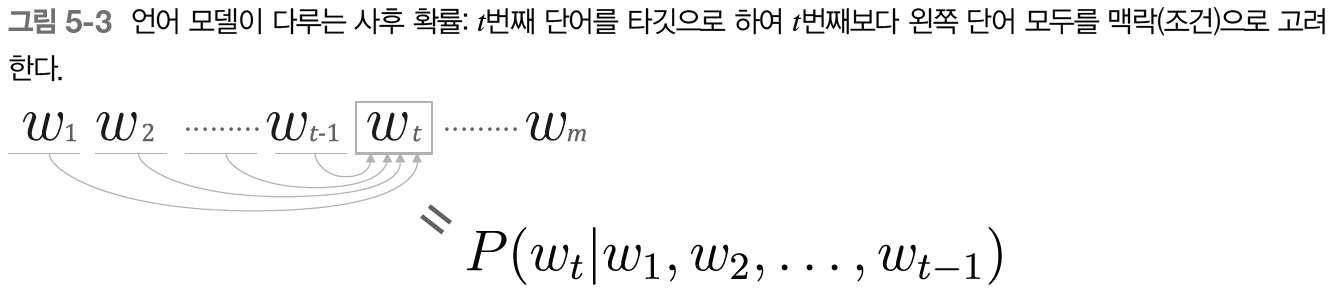
단어가 순서로 출현할 확률

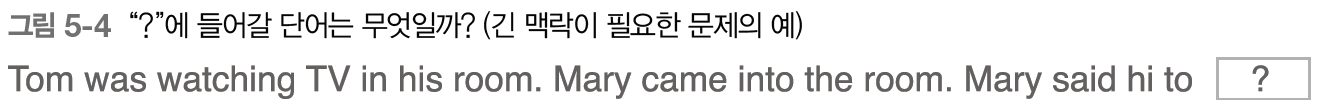


유도는 곱셈 공식 이용

C:\Users\smchiks\Desktop\머신러닝\deep-learning-from-scratch-2-master\equations_and_figures_2\deep_learning_2_images\e 5-5.png







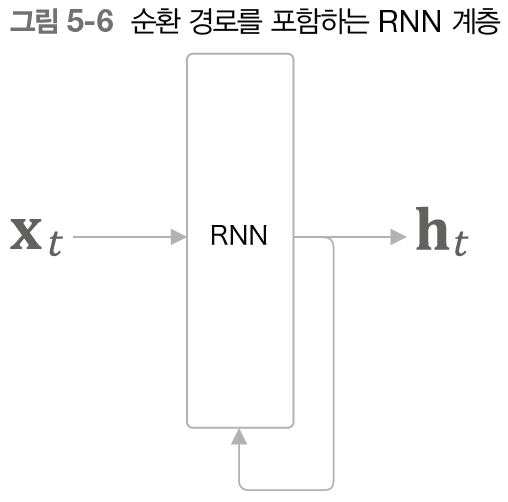
맥락의 크기가 커야 Tom이 예측가능, 큰 맥락은 계산 부담 🡪

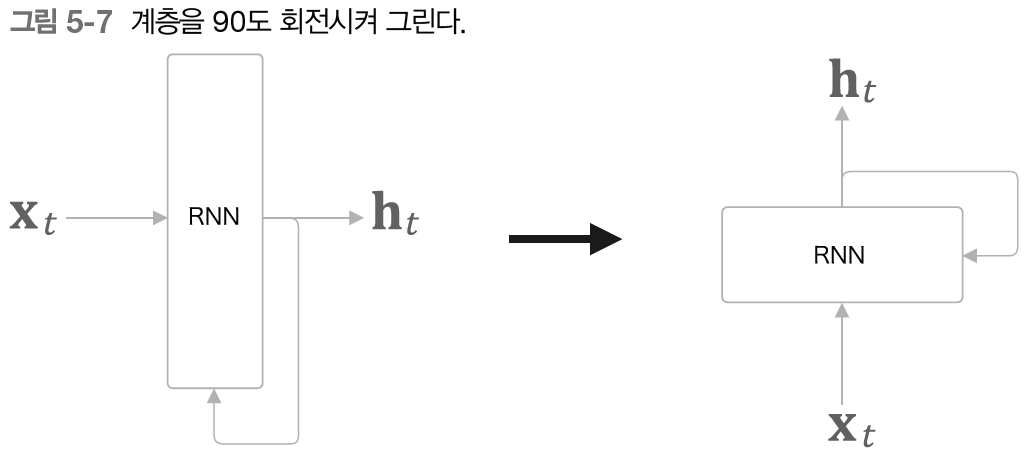
맥락 정보를 기억하는 mechanism 가진 RNN

5.2 RNN 이란

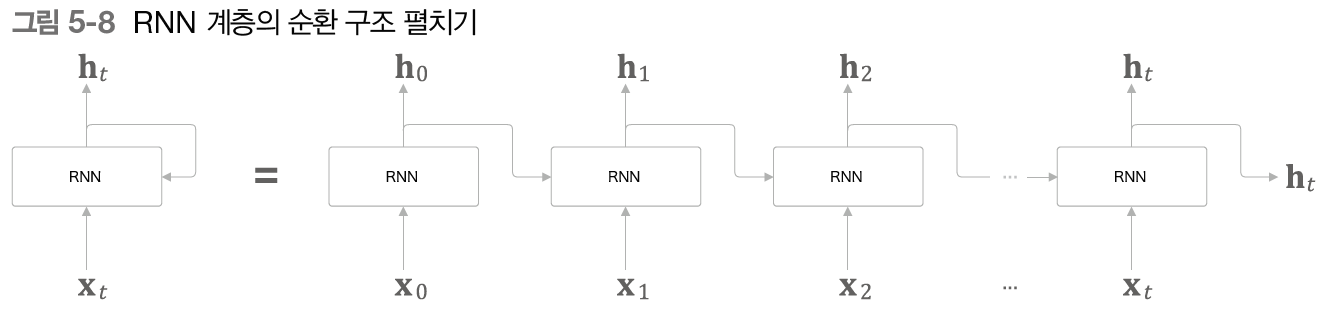
recurrent: 몇 번이나 반복해서 일어나는, 순환하는 신경망 🡪

은닉벡터가 시간에 따라 입력됨





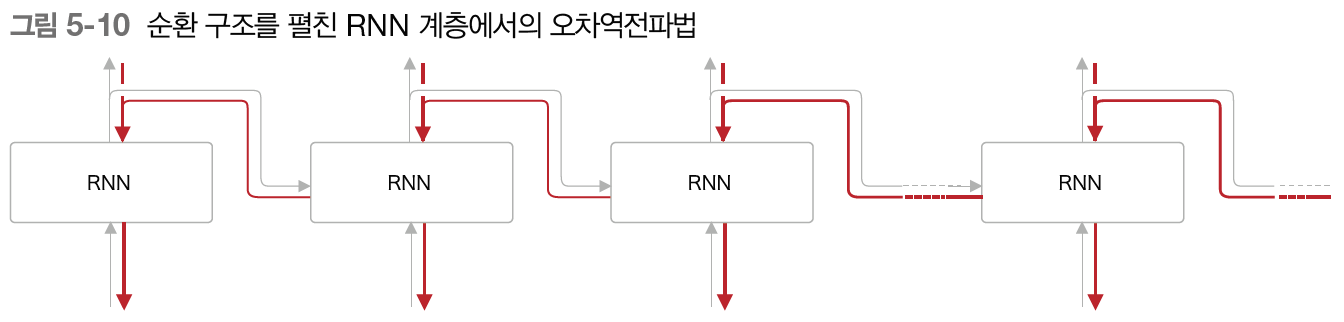
5.2.2 순환구조 펼치기



C:\Users\smchiks\Desktop\머신러닝\deep-learning-from-scratch-2-master\equations_and_figures_2\deep_learning_2_images\e 5-9.png

를 은닉 상태 hidden state, 은닉 상태 벡터라 함

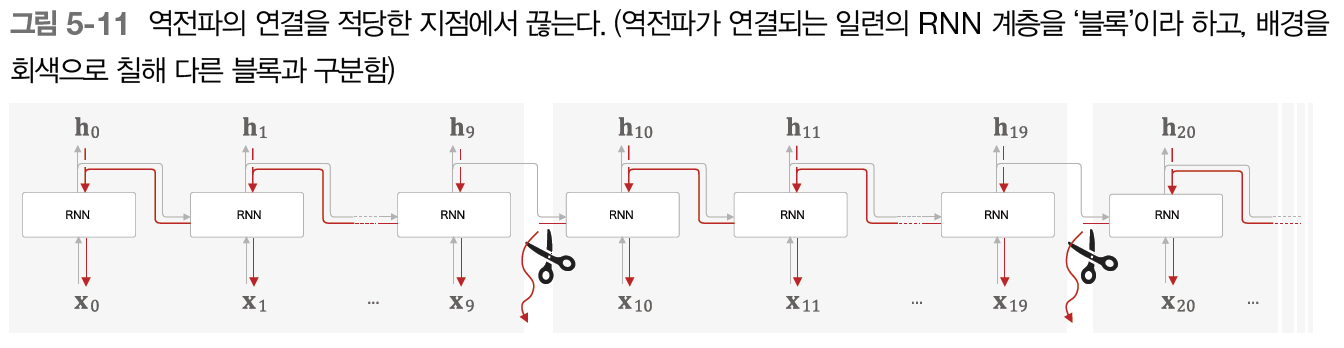
5.2.3 BPTT ( backpropagation through time)



시간 방향으로 펼친 신경망의 오차역전파법

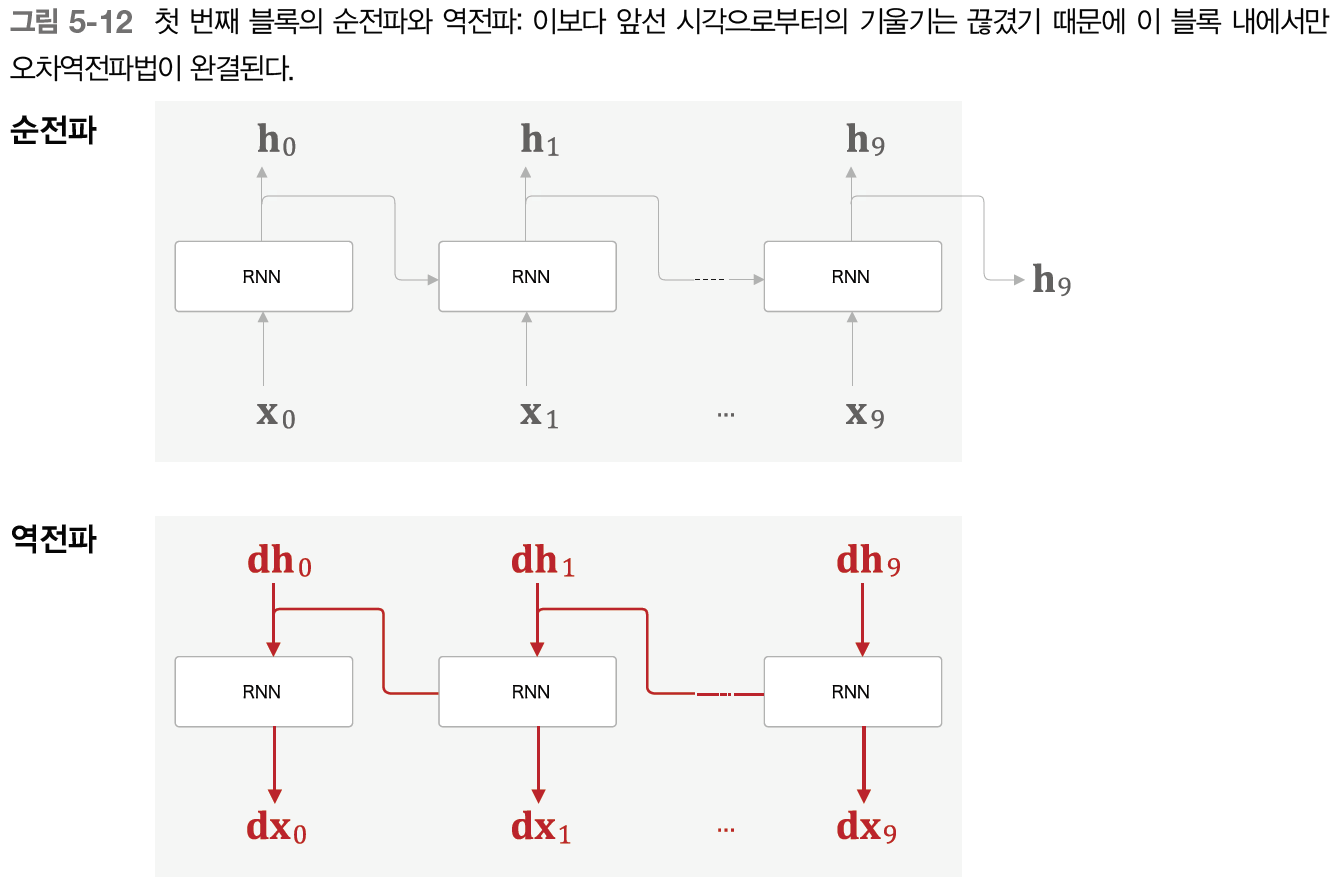
5.2.4 Truncated BPTT : 적당한 길이로 자름

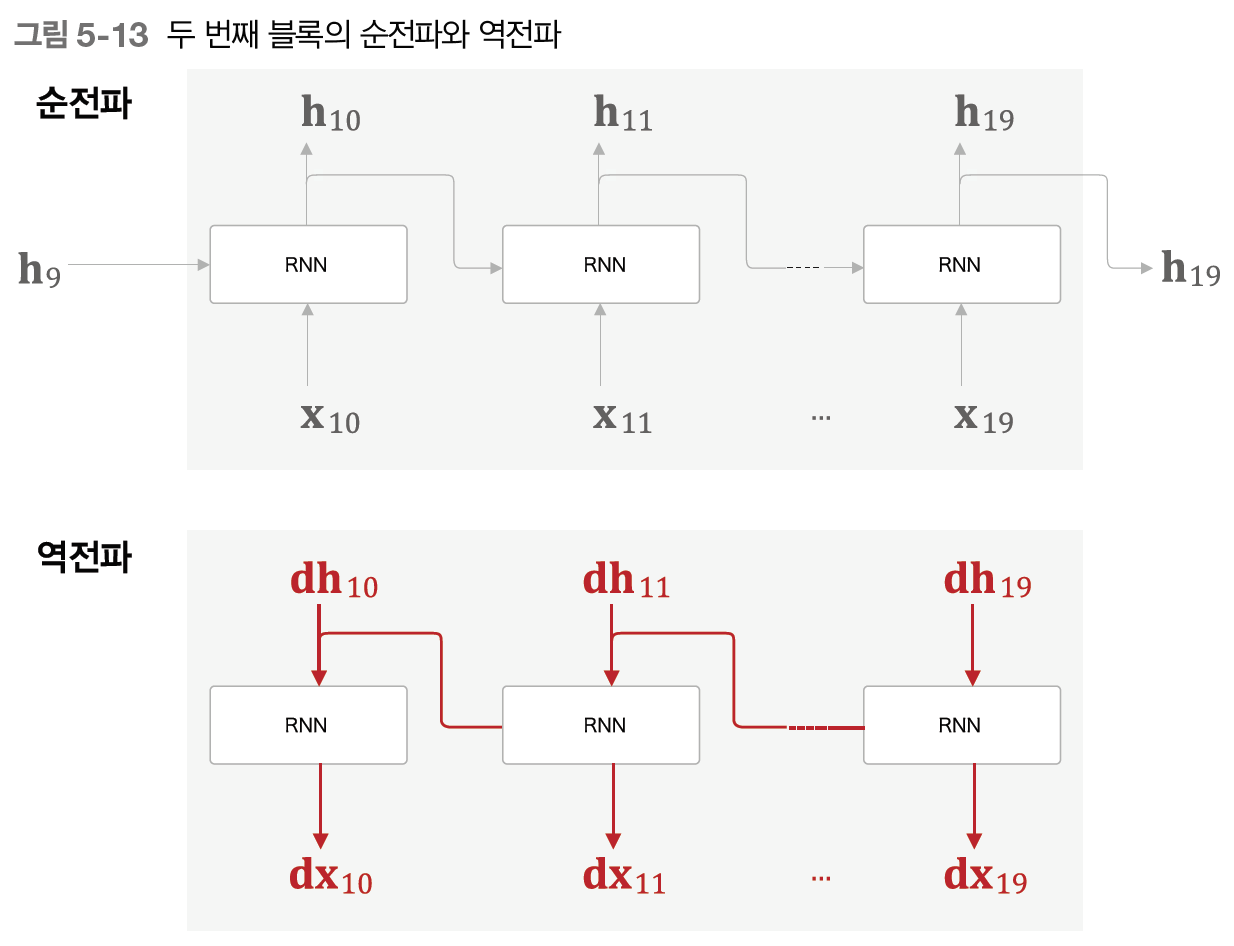
예) 길이가 10,000인 시계열은 계산량과 메모리 사용이 과다

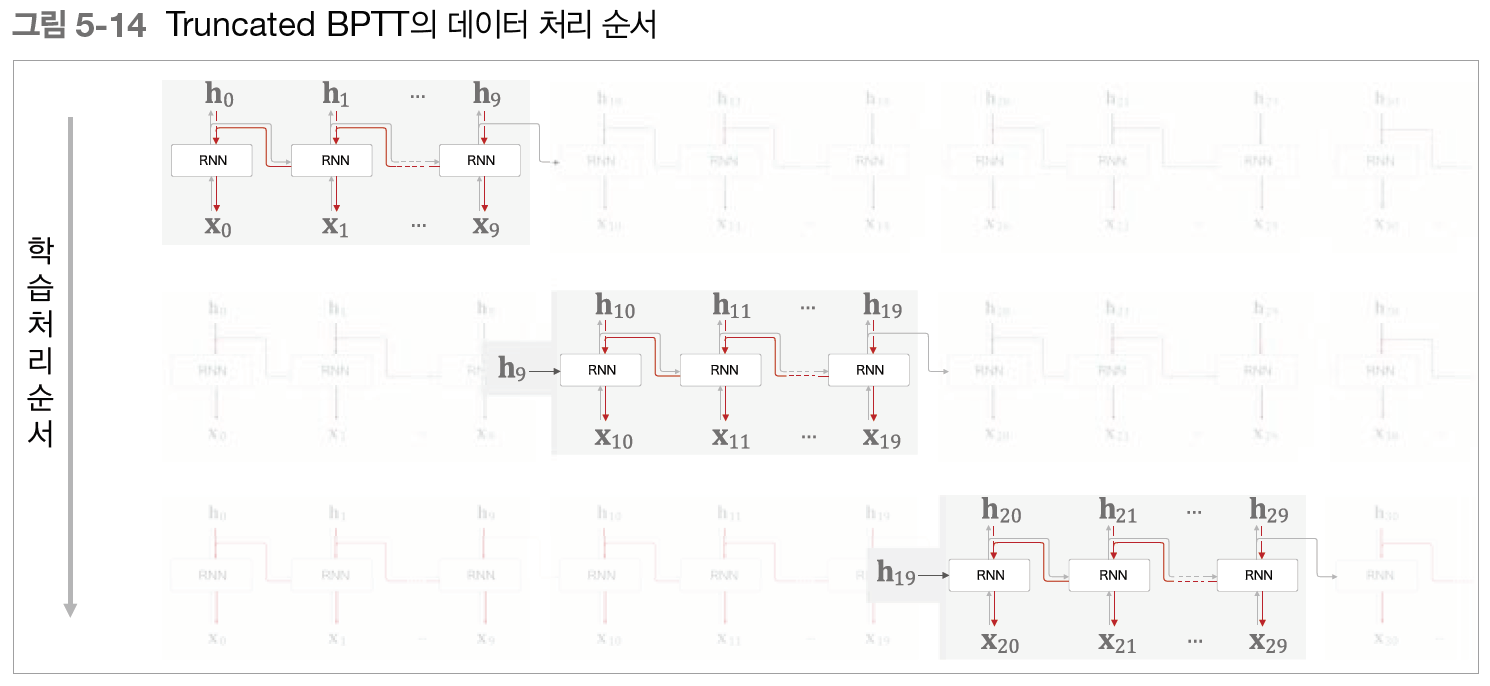


각각의 블록이 독립적으로 오차역전파를 완결

순전파의 연결은 유지



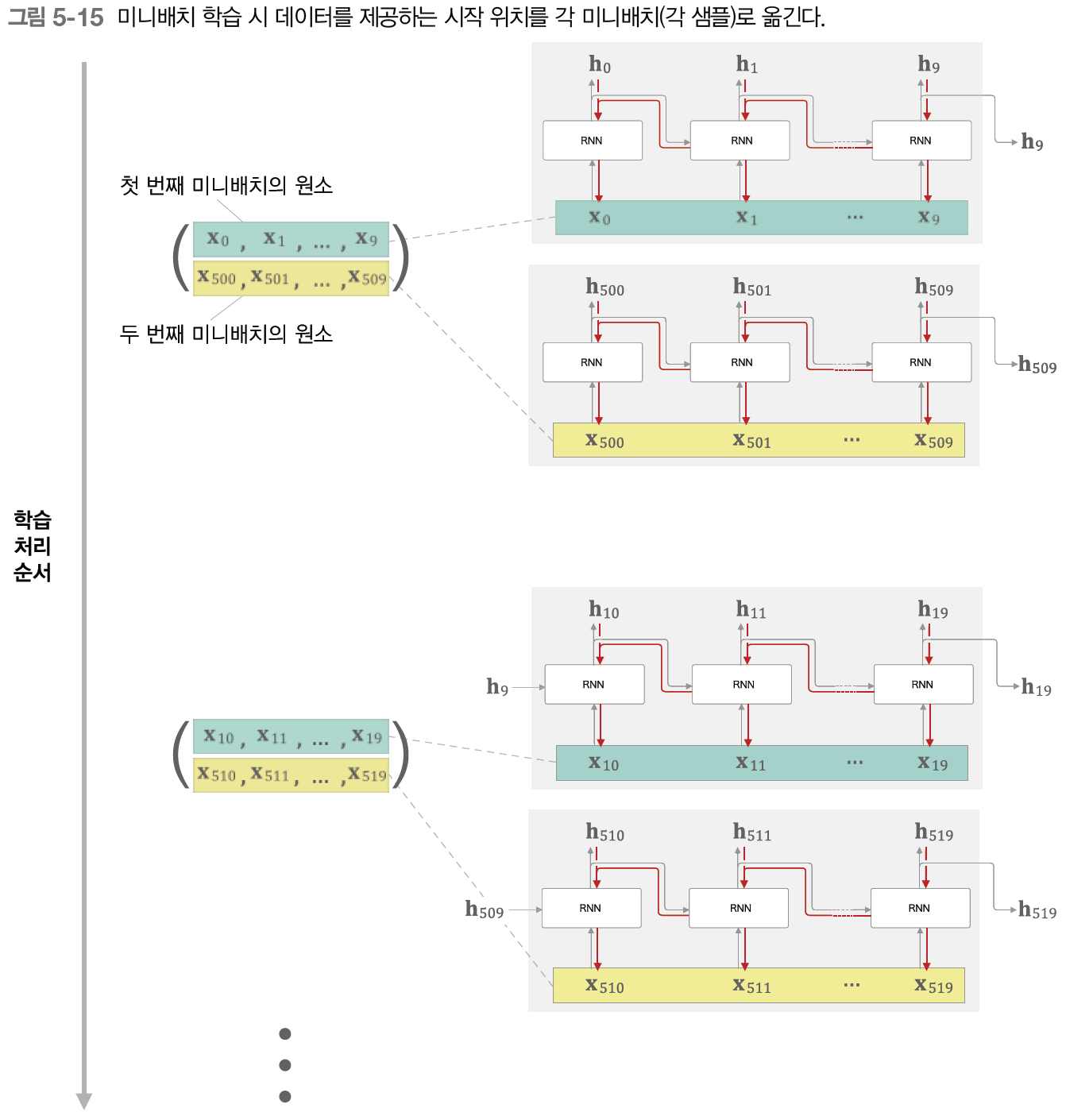




5.2.5 Truncated BPTT 의 미니배치 학습

데이터를 순서대로 제공하기

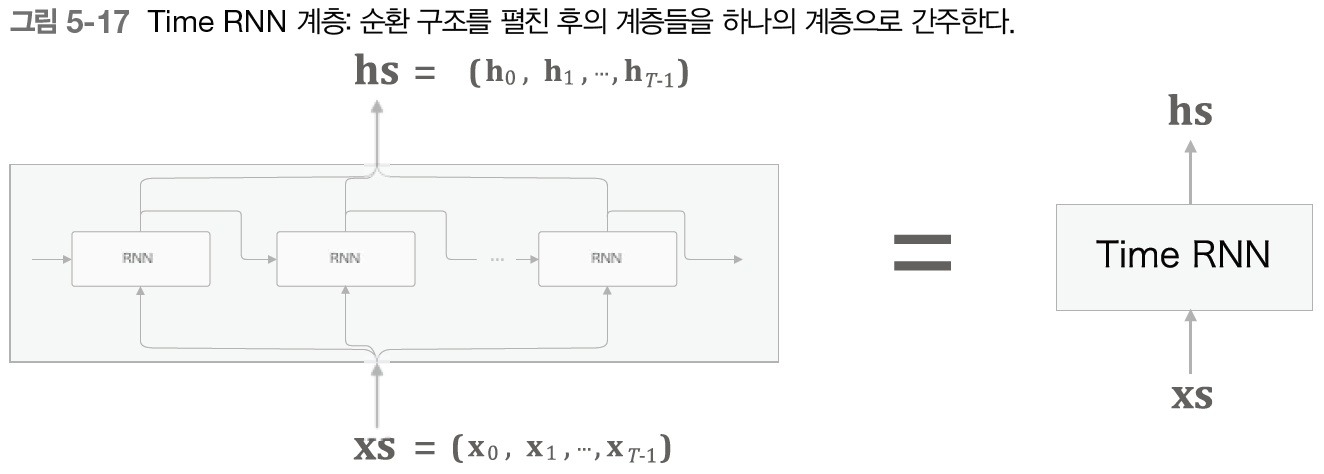
미니배치별로 데이터를 제공하는 시작 위치를 옮기기



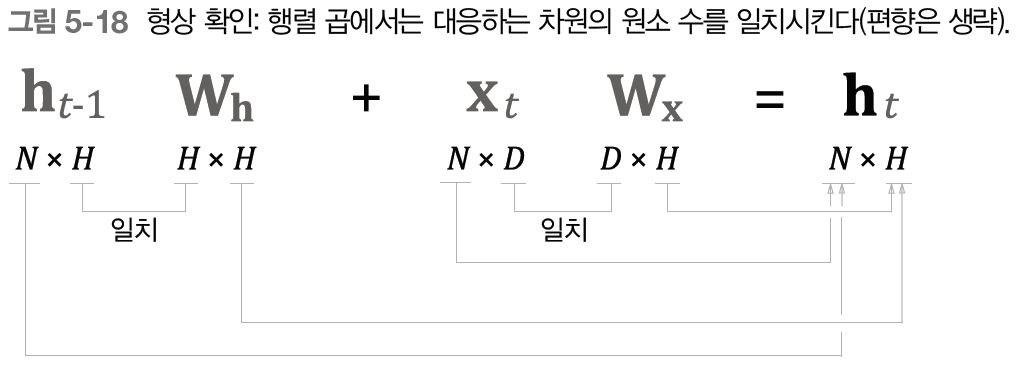
5.3 RNN 구현

한 단계의 작업을 수행하는 계층을 RNN 계층

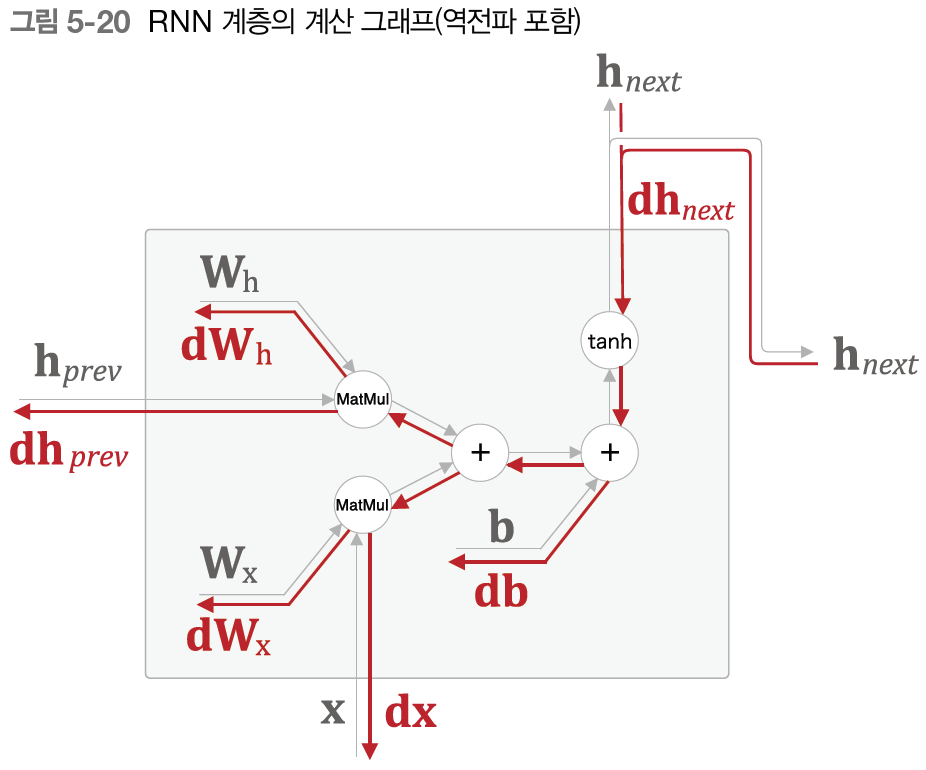
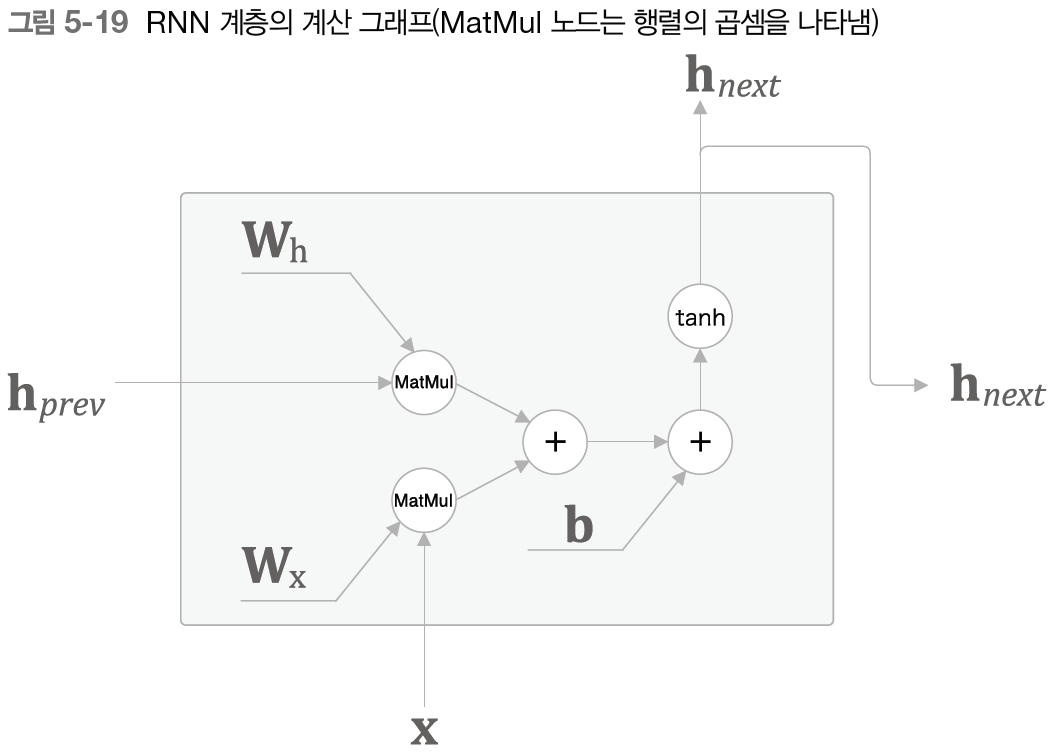
T개 단계분의 작업을 한꺼번에 처리하는 계층을 Time RNN 계층



5.3.1 RNN 계층 구현



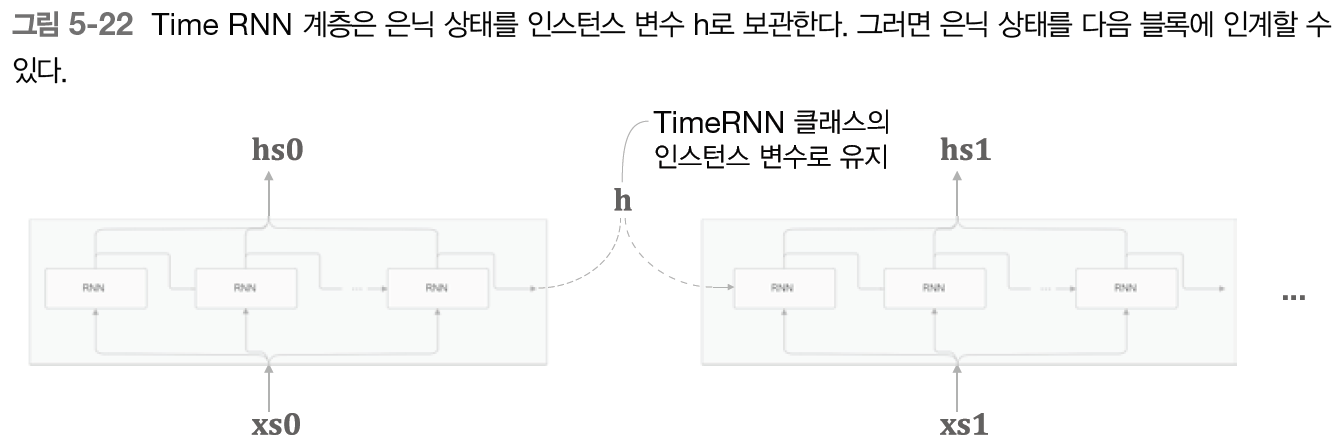
common/time\_layers.py 의 class RNN, forward 읽기



common/time\_layers.py 의 class RNN, backward 완벽 이해?

5.3.2 Time RNN 계층 구현

T개의 RNN계층을 연결한 신경망

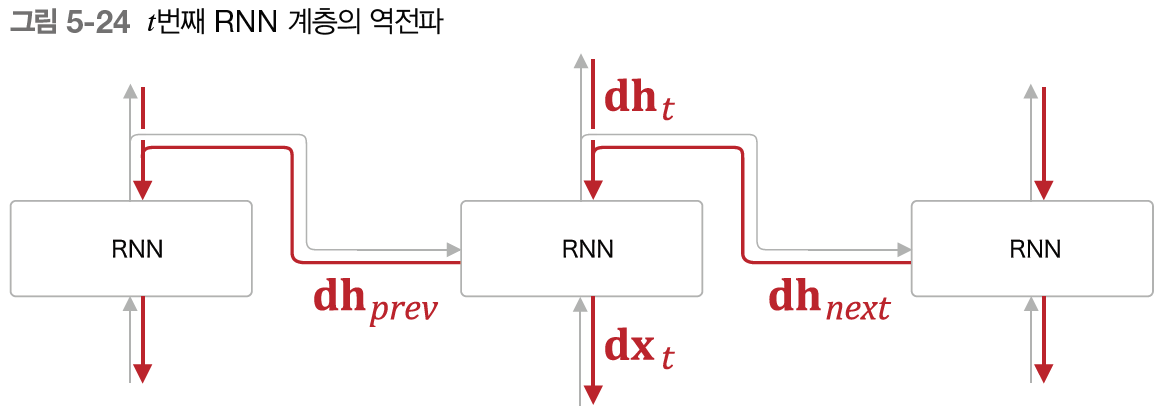


common/time\_layers.py 의 class TimeRNN

\_\_init\_\_ : h, layers 변수

forward : np.empty 초기화되지 않은 배열 반환, layers: RNN 들 보관

backward: 분기되었던 것을 합침 (합의 미분)



RNN 계층이 똑 같은 가중치를 사용하므로, Time RNN 기울기는 모든 기울기의 합, 즉

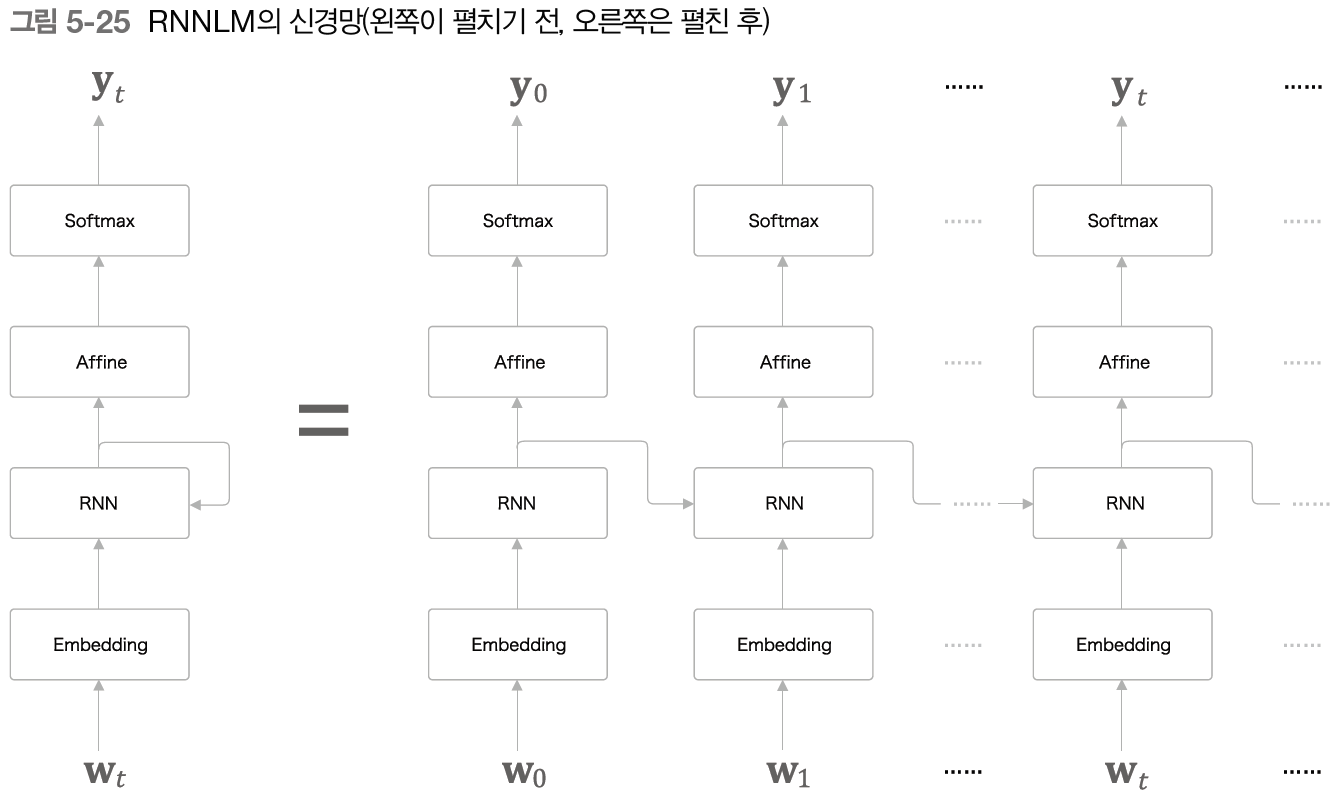
for i, grad in enumerate(layer.grads):

grads[i] += grad

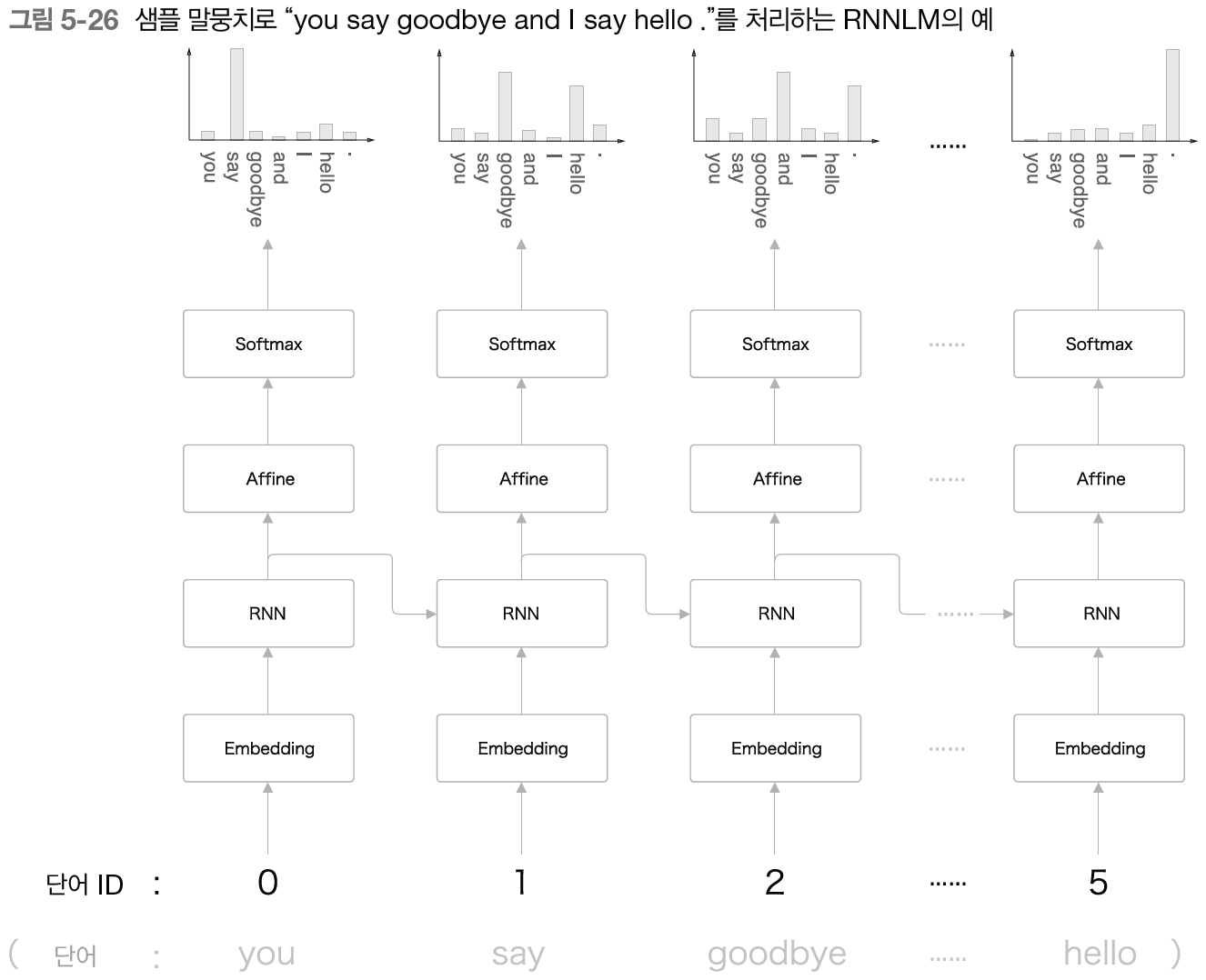
5.4 시계열 데이터 처리 계층 구현

RNN의 여러 응용 중의 하나로서

2,3,4장에서 배운 언어 모델을 구현



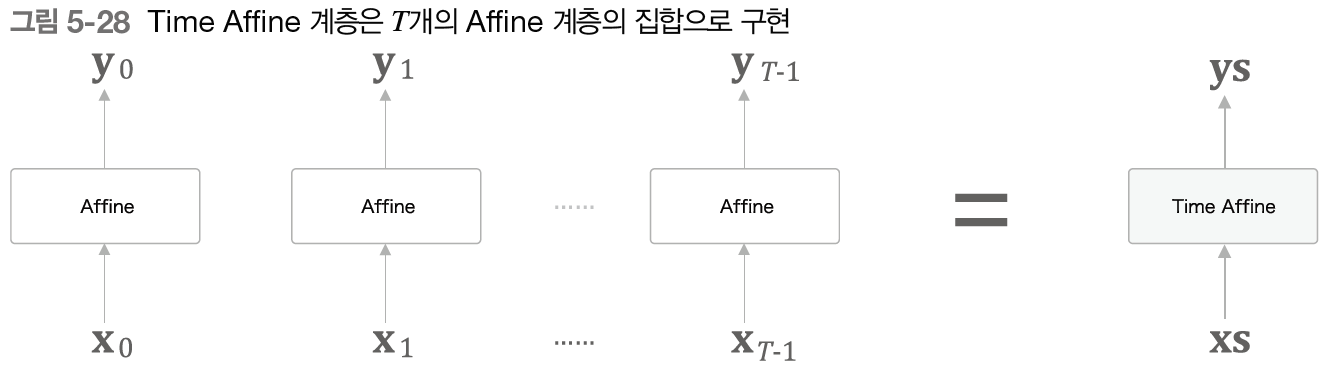
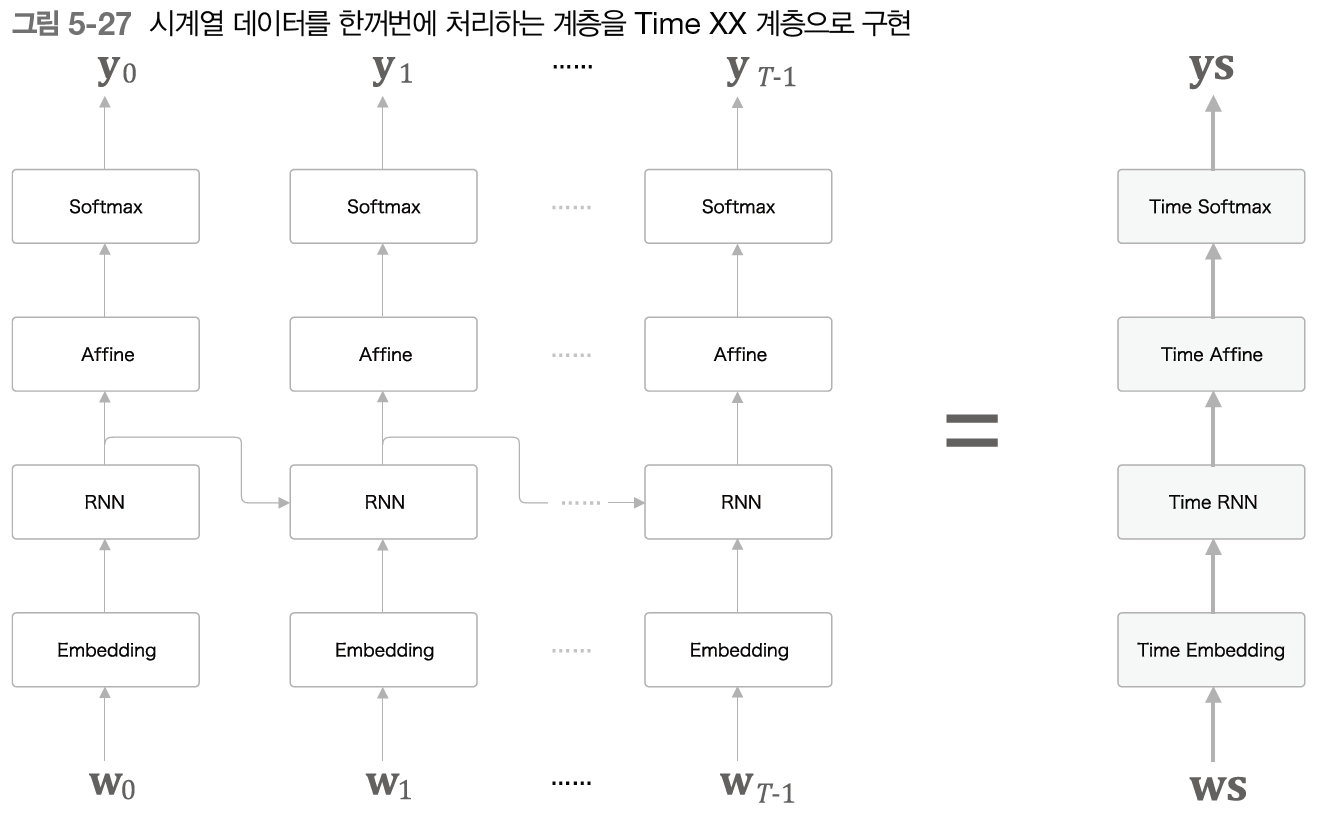
Embedding 계층: 단어의 분산표현



RNNLM 지금까지의 입력된 단어를 기억하고 다음에 출현할 단어를 예측 🡨 RNN 계층이 과거 정보를 인코딩해 저장 해줌

5.4.2 Time 계층 구현

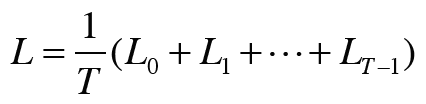
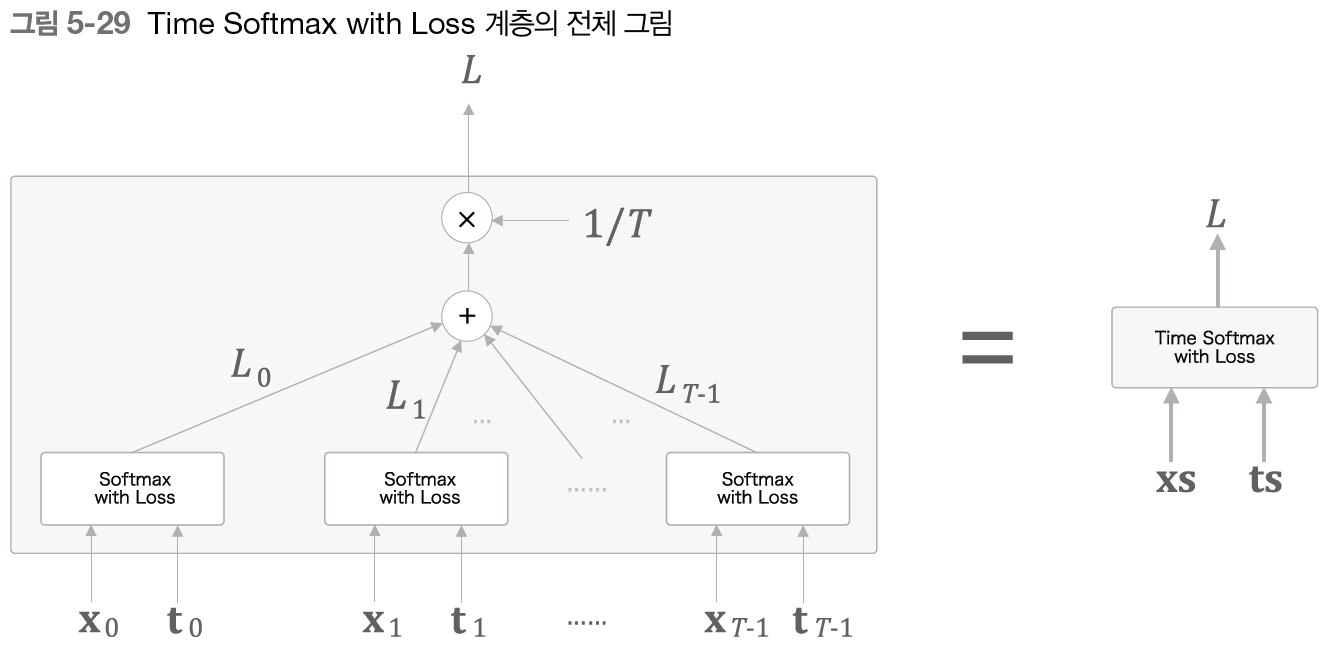
Time Embedding, Time Affine



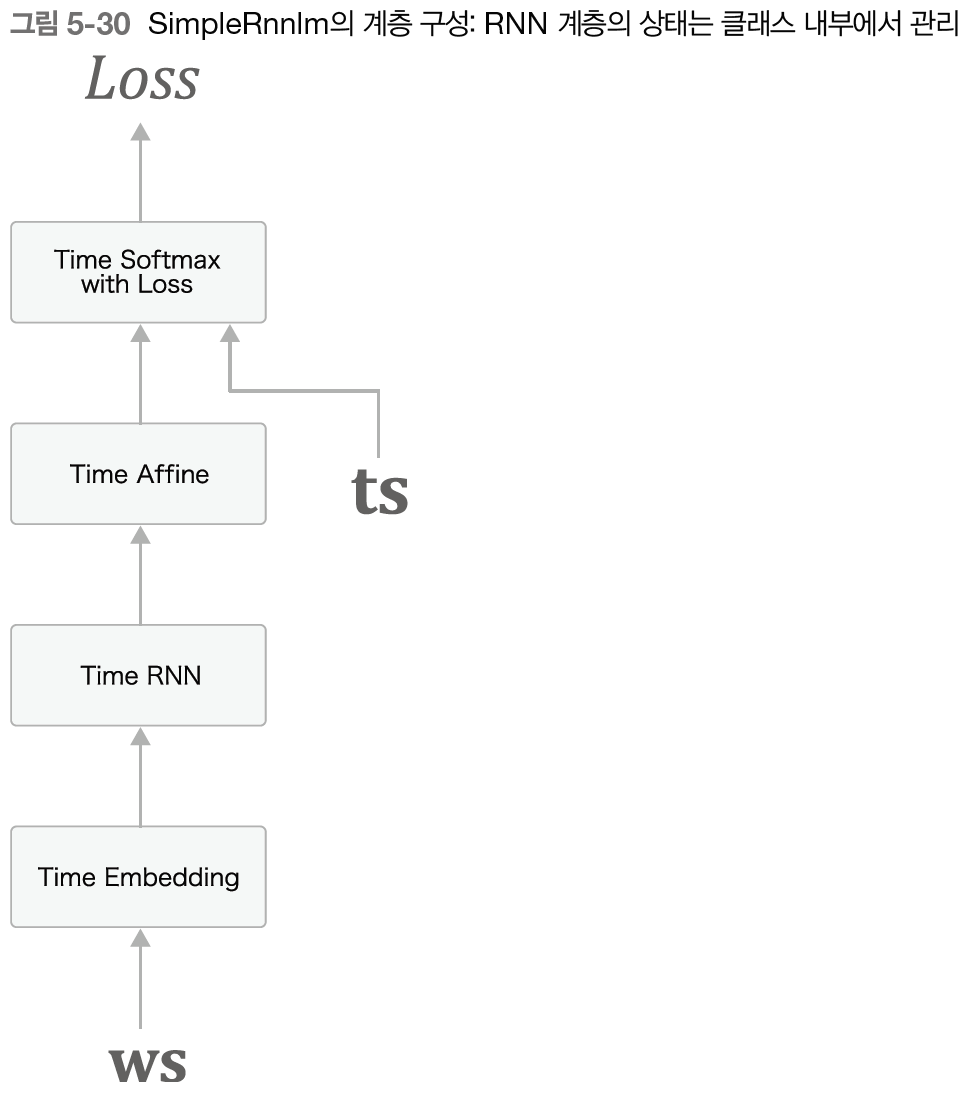
Time Affine, Time Embedding은 각각 T개 계층을 시간별로 처리 🡪

독립적이므로 한꺼번에 처리 가능

마찬가지로



5.5 RNNLM 학습과 평가



ch05/simple\_rnnlm.py 읽기

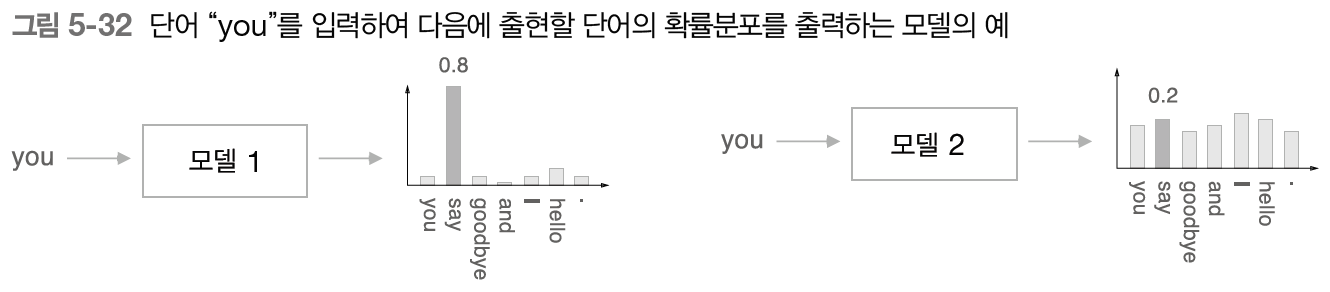
\_\_init\_\_ : 가중치 초기화, Xavier 초기화

forward(), backward(), reset\_state() 를 차례로 수행

5.5.2 언어 모델의 평가

언어모델은 주어진 단어로부터 다음에 출현할 단어 예측

perplexity : 혼란도, 언어모델의 예측 성능을 평가, 확률의 역수로 정의되므로 작을수록 정확, 분기수



log y를 log p로 생각하면 확률 역수 의미가 됨



5.5.3 RNNLM의 학습 코드

ch05/train\_custom\_loop.py 읽기

data\_size = len(xs) : 1개 제외 (마지막 단어는 target 없음)

# 미니배치의 각 샘플의 읽기 시작 위치를 계산

jump = (corpus\_size - 1) // batch\_size #1개 제외, 999 // 10 🡪 99

time\_idx 가 계속 증가하면서 전체 자료를 여러 번 읽음, % data\_size로 전체 자료 길이를 초과하지 않음

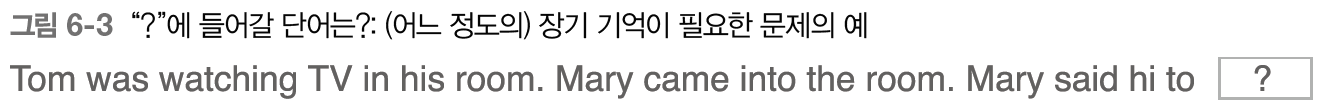
chapter 6 게이트가 추가된 RNN

RNN: 장기 의존 관계의 학습이 어려움

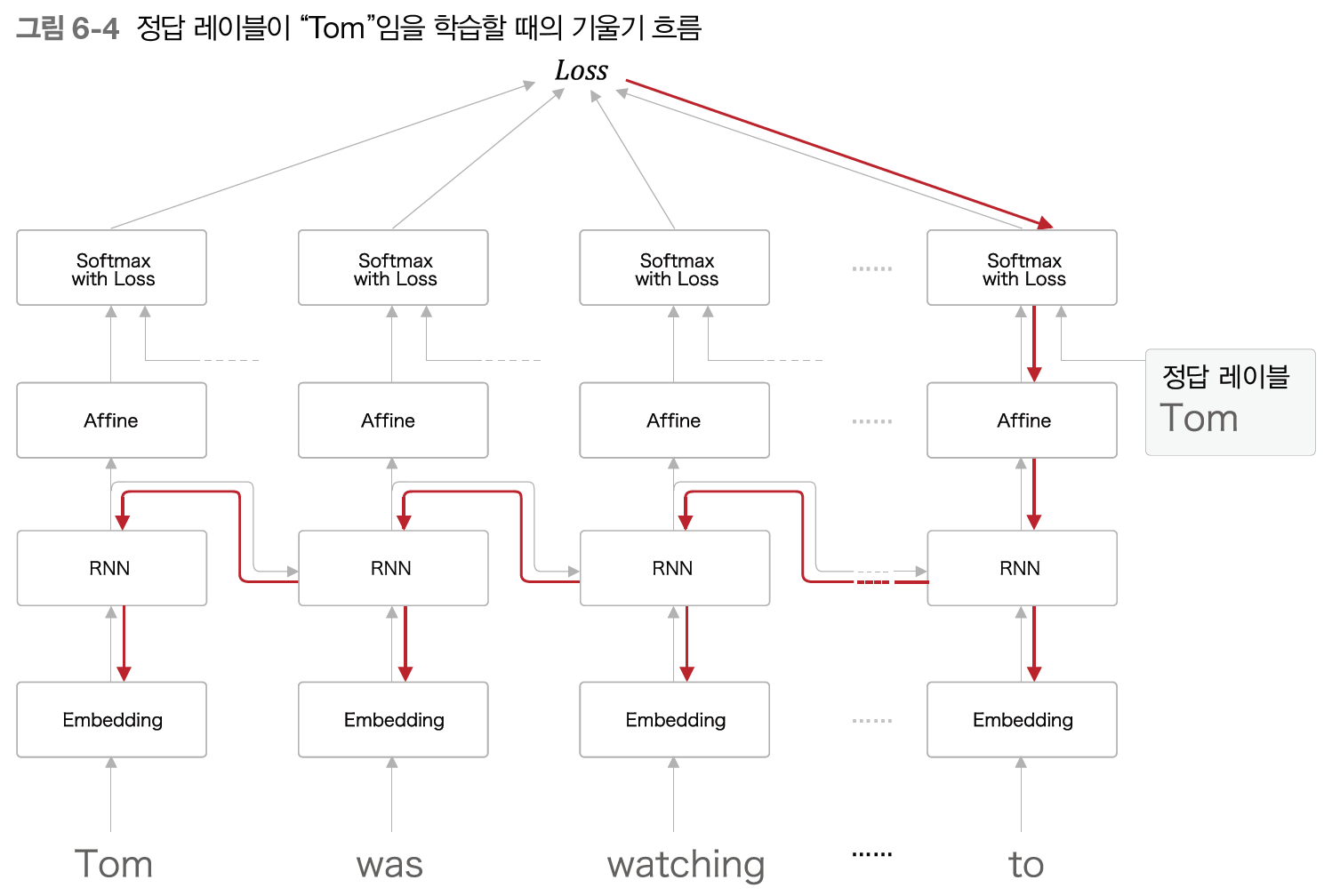
LSTM, GRU: gate를 추가하여 장기기억 가능하게 함

6.1 RNN의 문제점

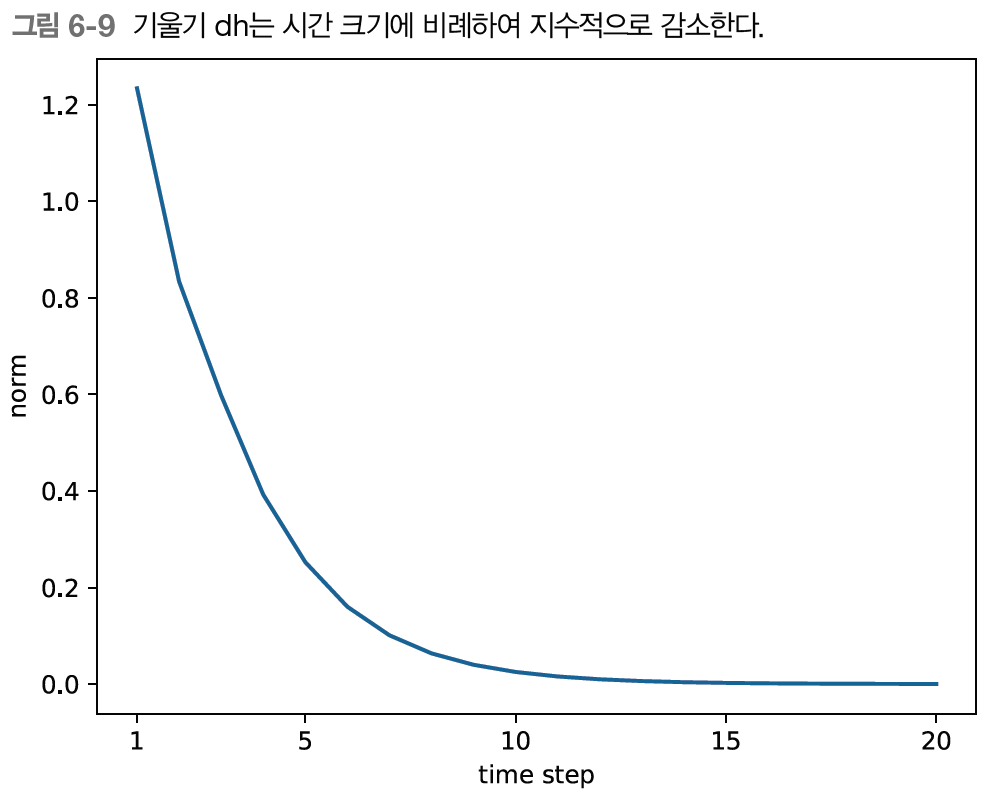
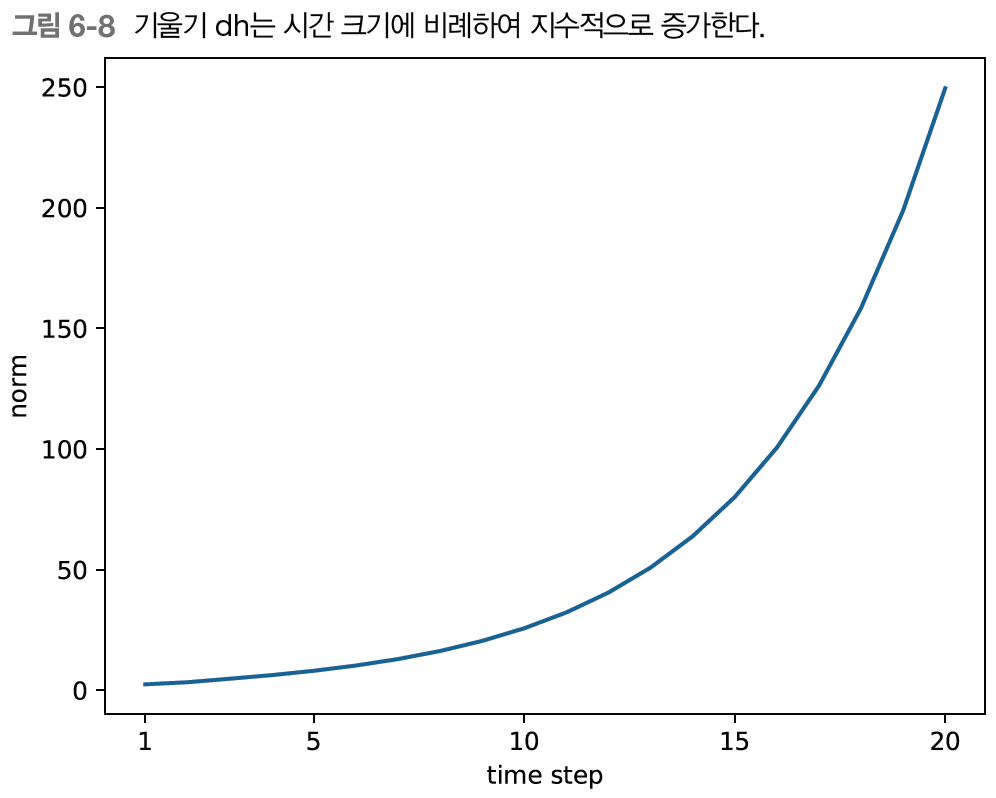
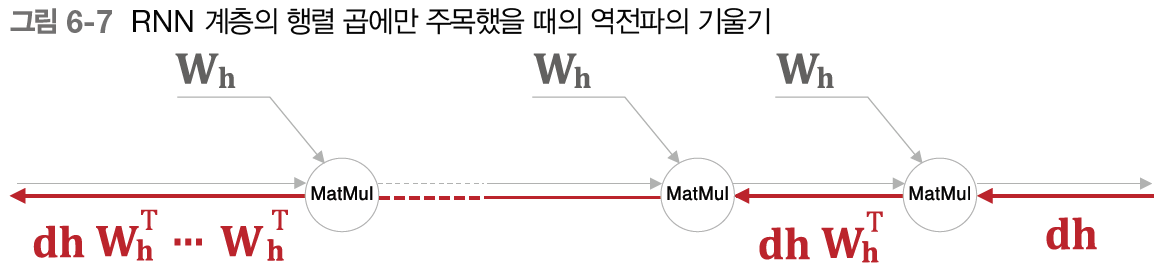
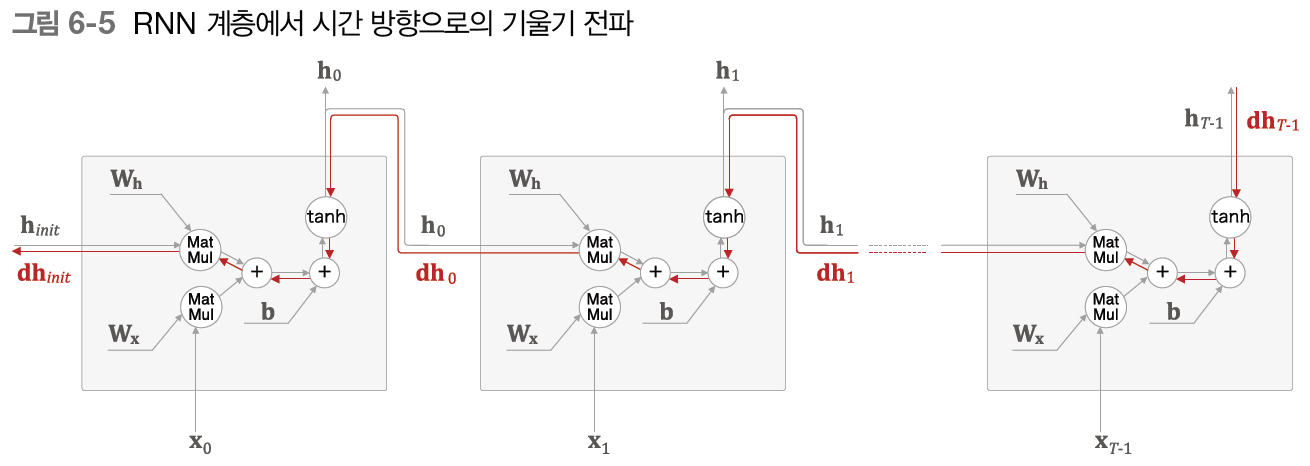
BPTT에서 기울기의 소실이나 폭발이 발생



정답 레이블이 사용되는 Loss로부터 멀다.



6.1.3 기울기 소실과 기울기 폭발의 원인



6.1.4 *기울기 폭발 대책*

*gradient clipping*

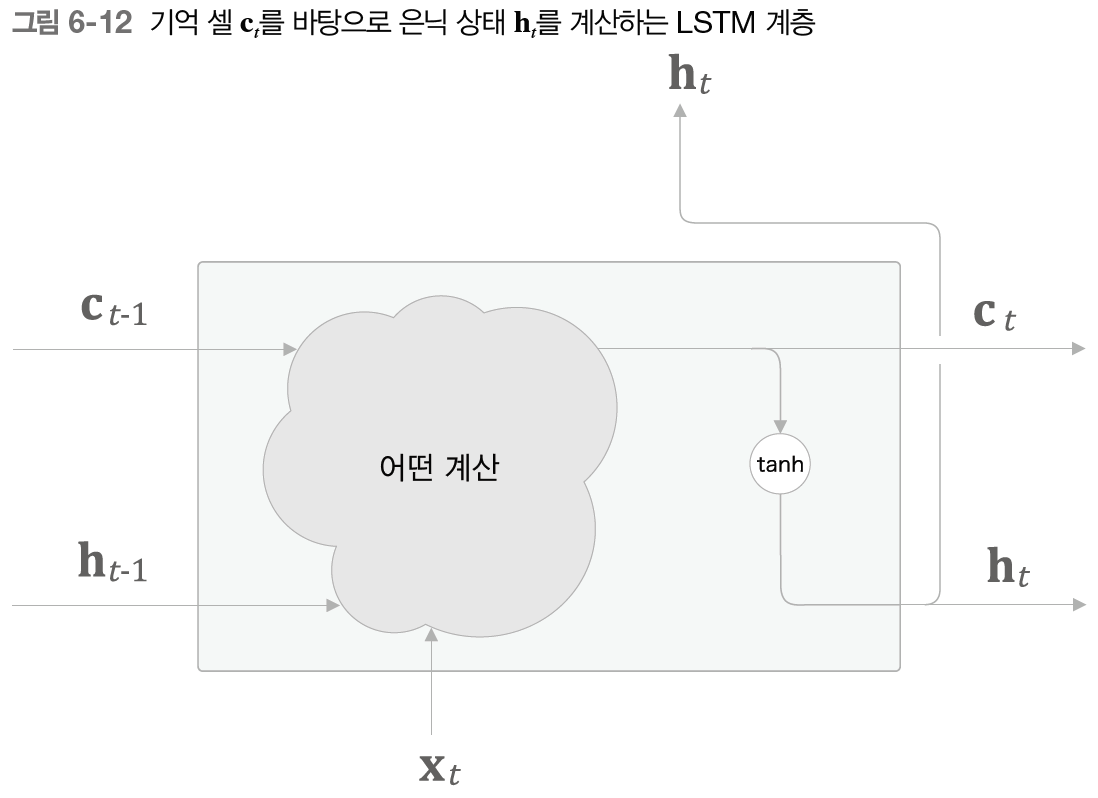
*common/util.py clip\_*grads(grads, max\_norm) 읽기

6.2 기울기 소실과 LSTM(Long Short-Term Memory 단기기억을 긴 시간 지속)

기울기 소실을 신경망 구조로서 완화시킴

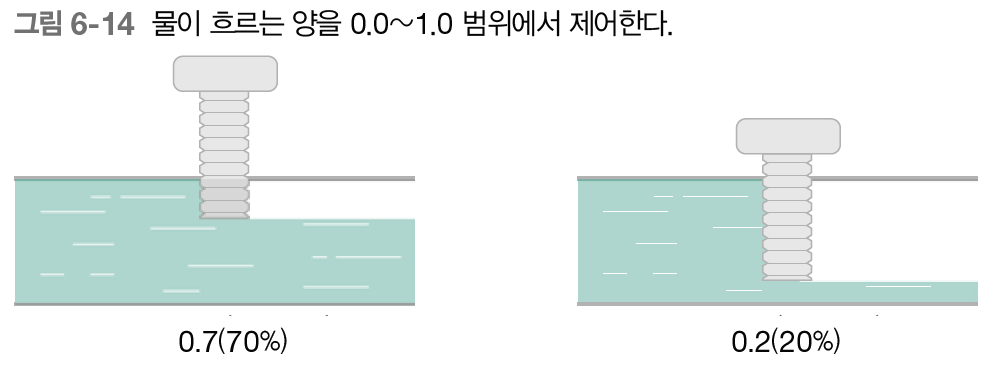
6.2.1 LSTM의 인터페이스

c: memory cell, LSTM내에서만 전달



이 구조를 gate를 사용하여 정교하게 구현

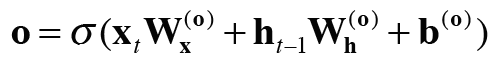
gate



gate를 여는 정도를 학습

6.2.3 output gate

: sigmoid 🡪 [0, 1]

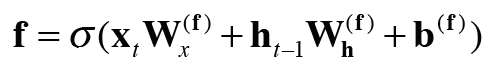
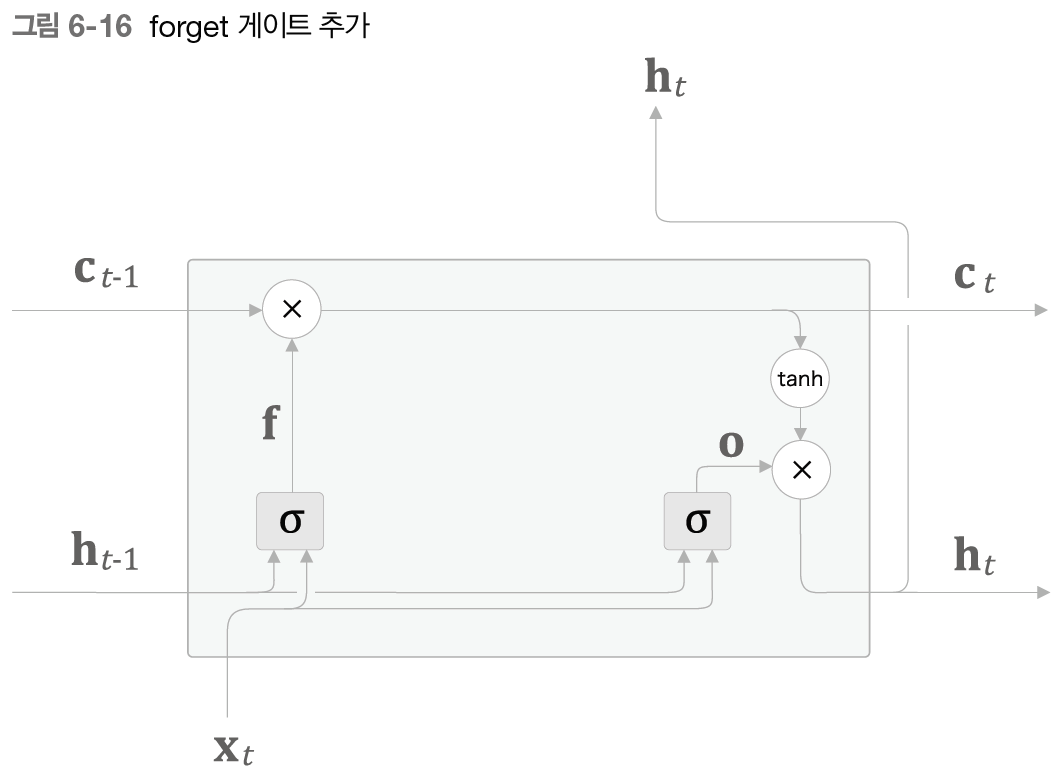


Hadamard product : 원소별 곱



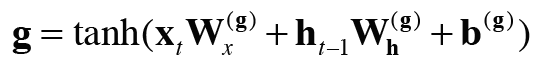
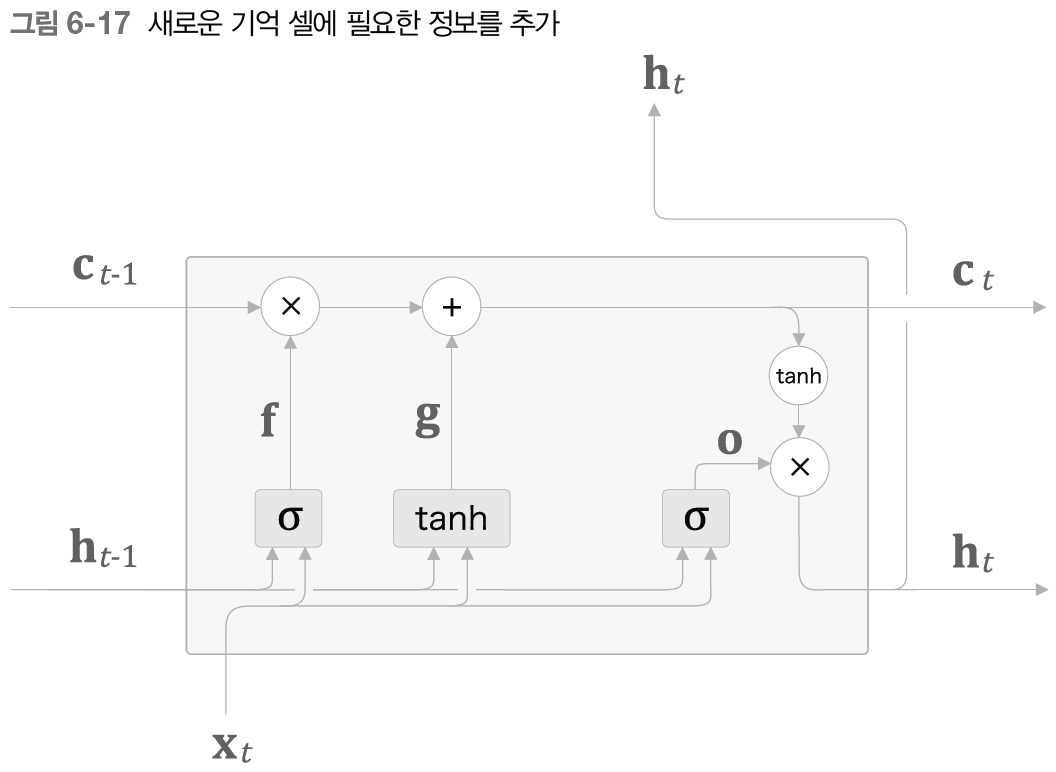
6.2.4 forget gate

불필요한 기억을 잊게 해주는 gate



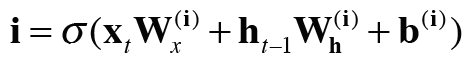
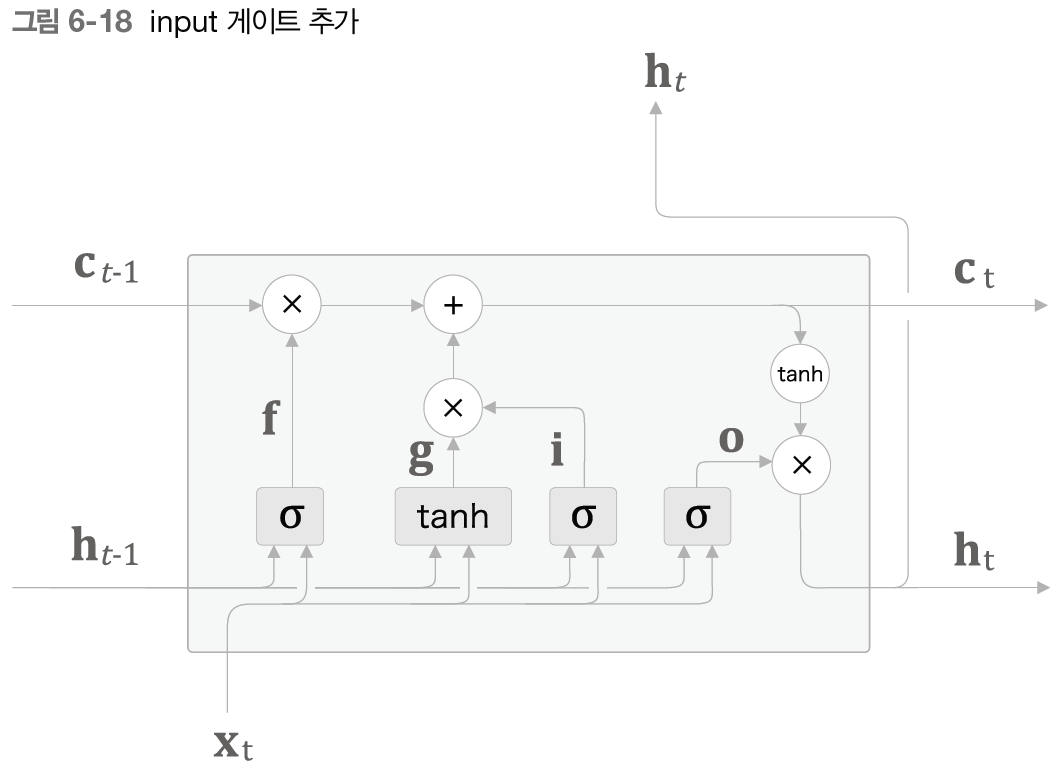
6.2.5 새로운 기억 셀

새로 기억해야할 정보를 추가

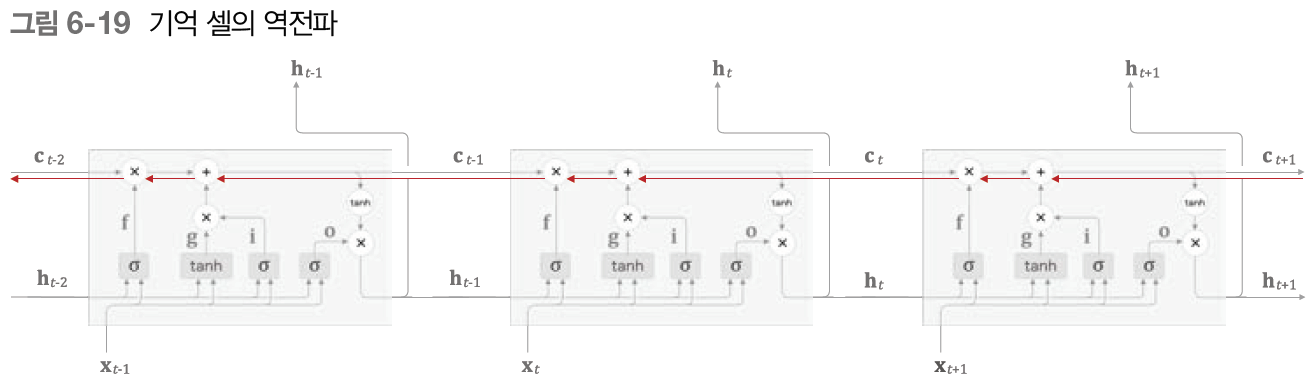


6.2.6 input gate

새로 추가되는 정보의 가치 판단



6.2.7 LSTM의 기울기 흐름



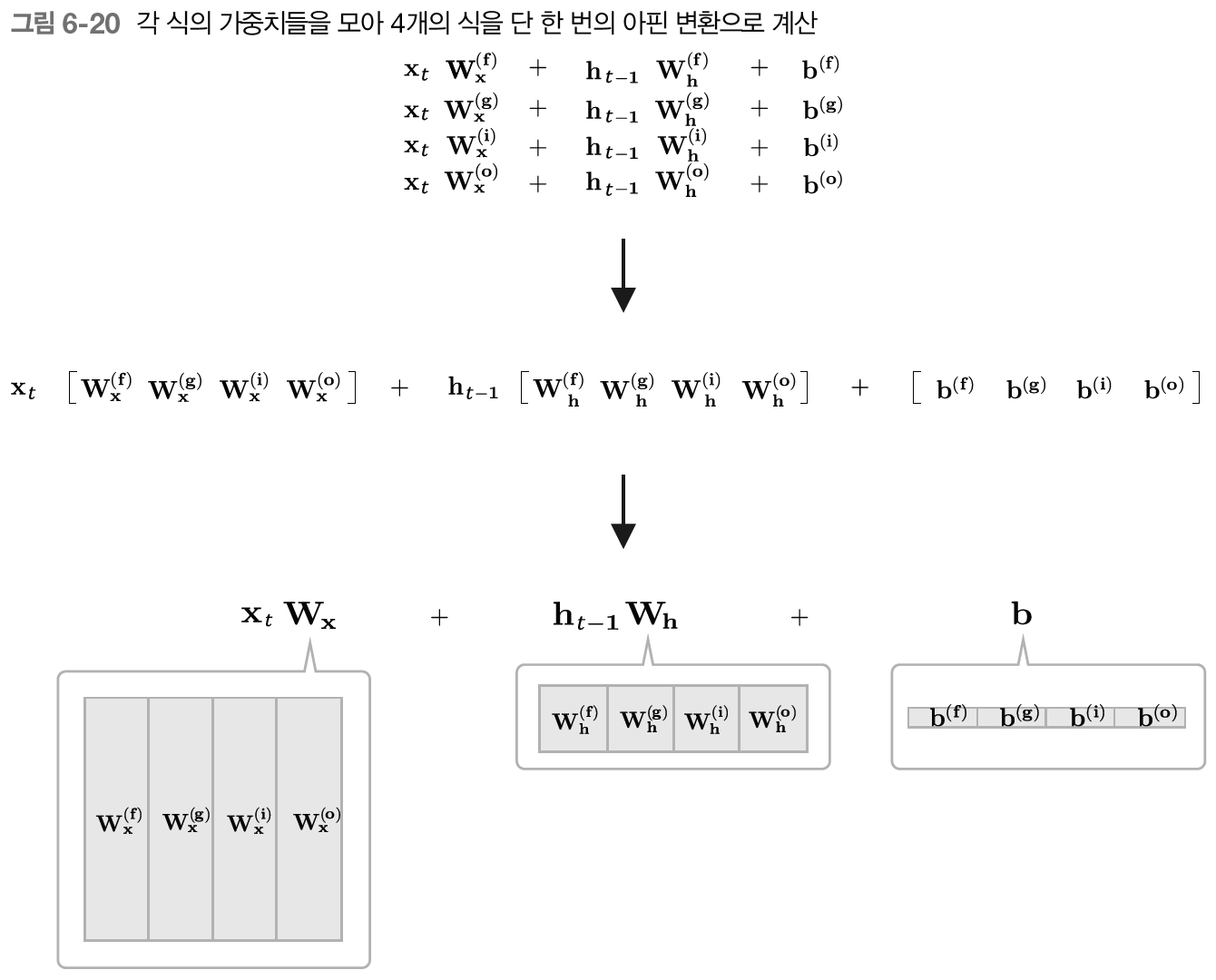
역전파시 +와 hadamard 곱 통과

+는 기울기가 그대로 전달

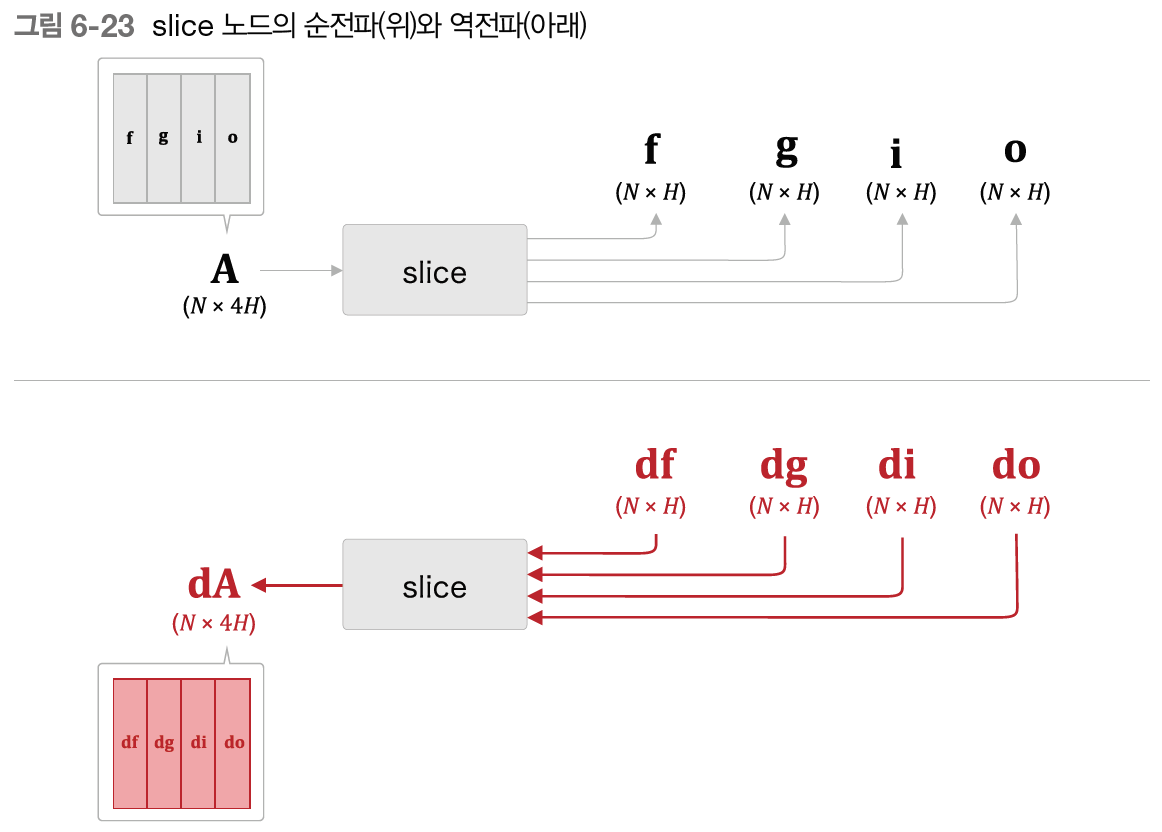
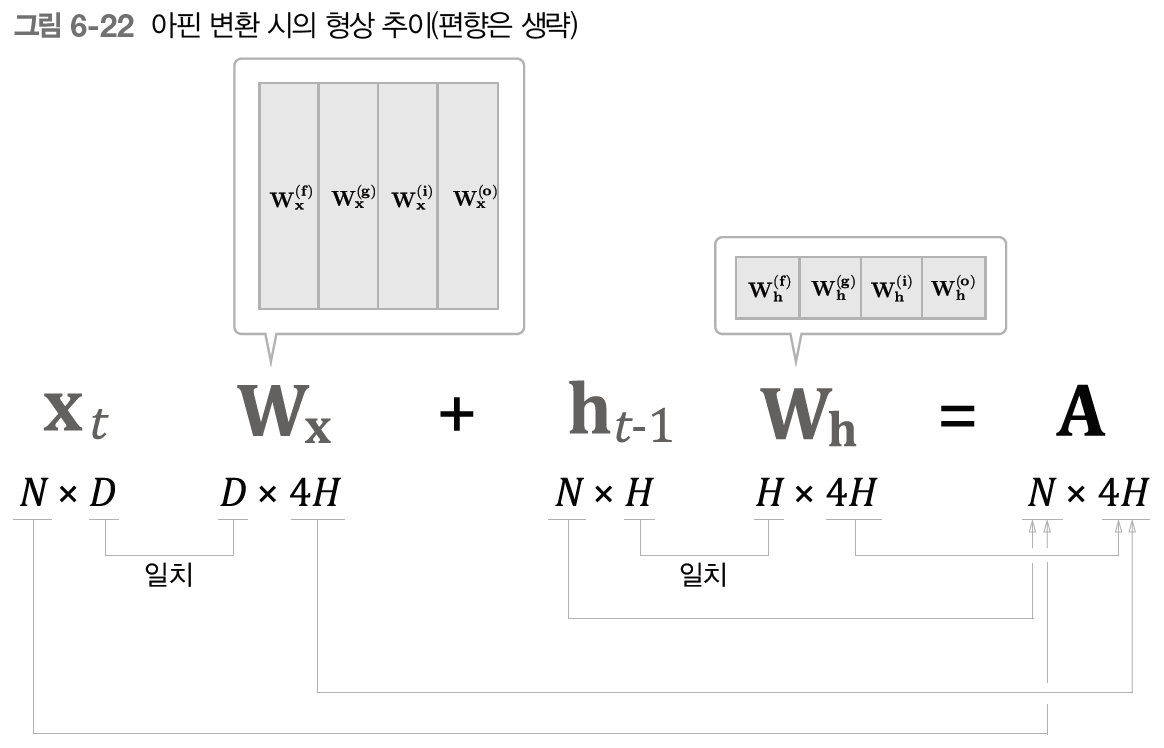
원소별 곱은 잊어야 하는 원소에 대해서만 작은 값이 곱해짐

즉, 기울기 소실이 완화되는 역전파 구조임

6.3 LSTM 구현



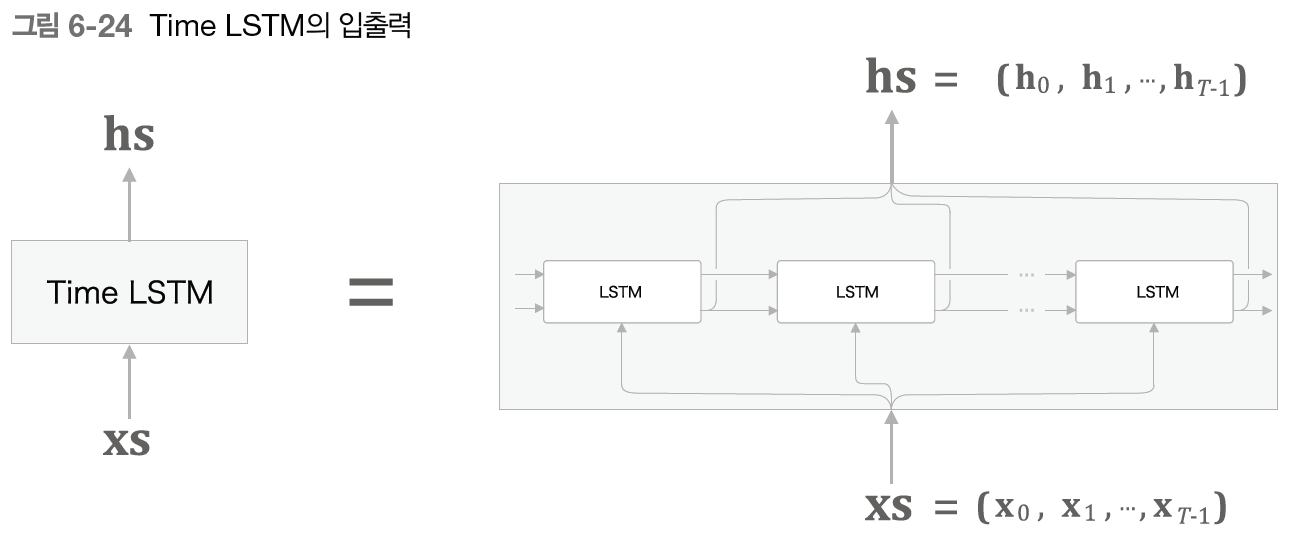
common/time\_layers.py의 class LSTM 읽기

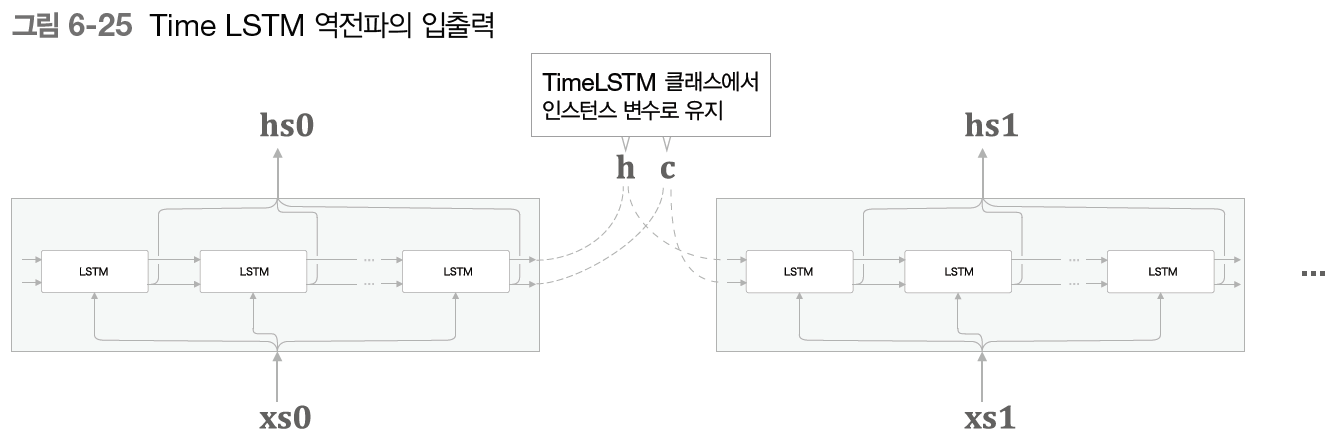


backward(): ds는 c\_t-1과 c\_t 사이의 + 기호 다음의 gradient

6.3.1 Time LSTM 구현

common/time\_layers.py의 class TimeLSTM 읽기

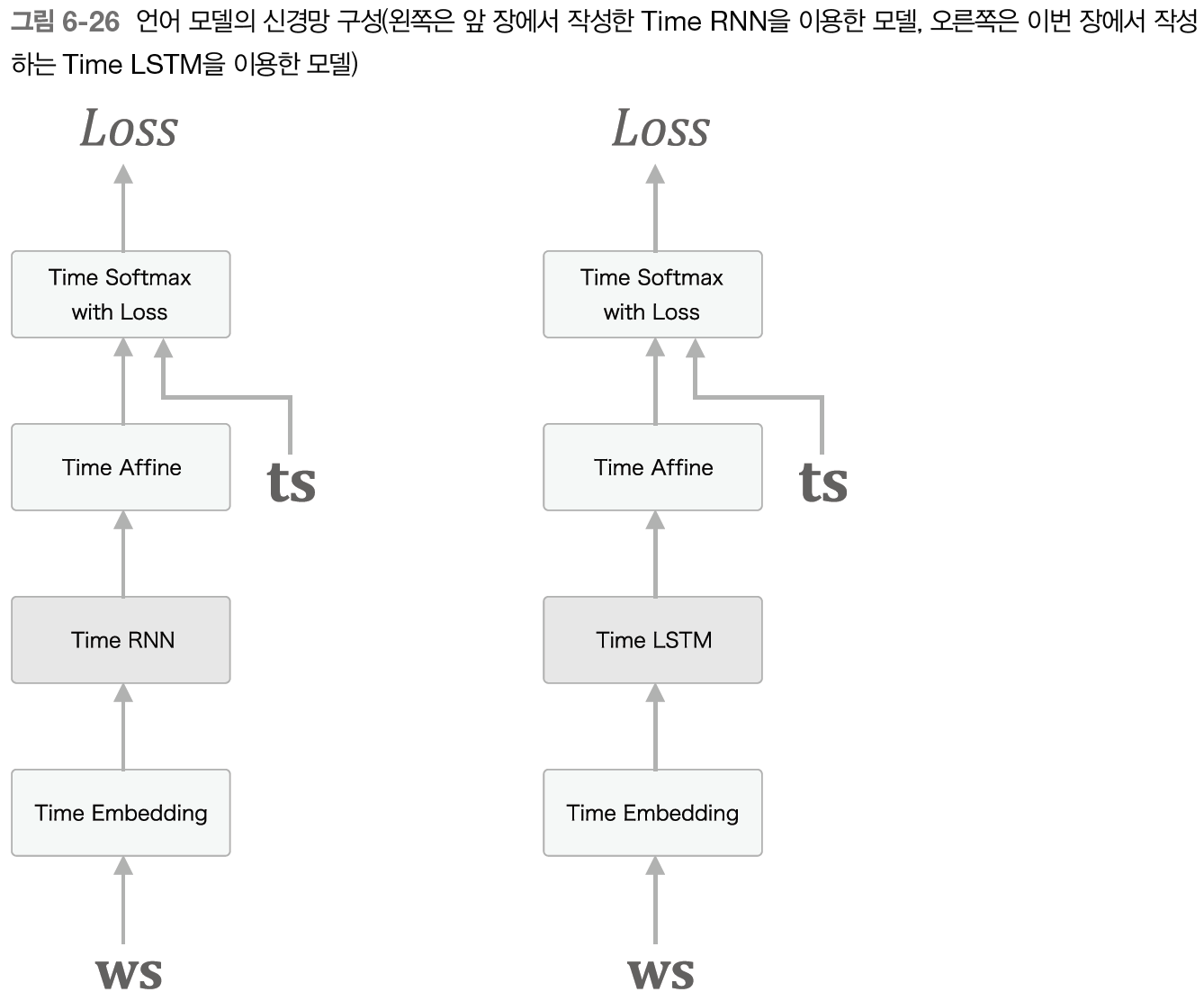




마지막에 h, c

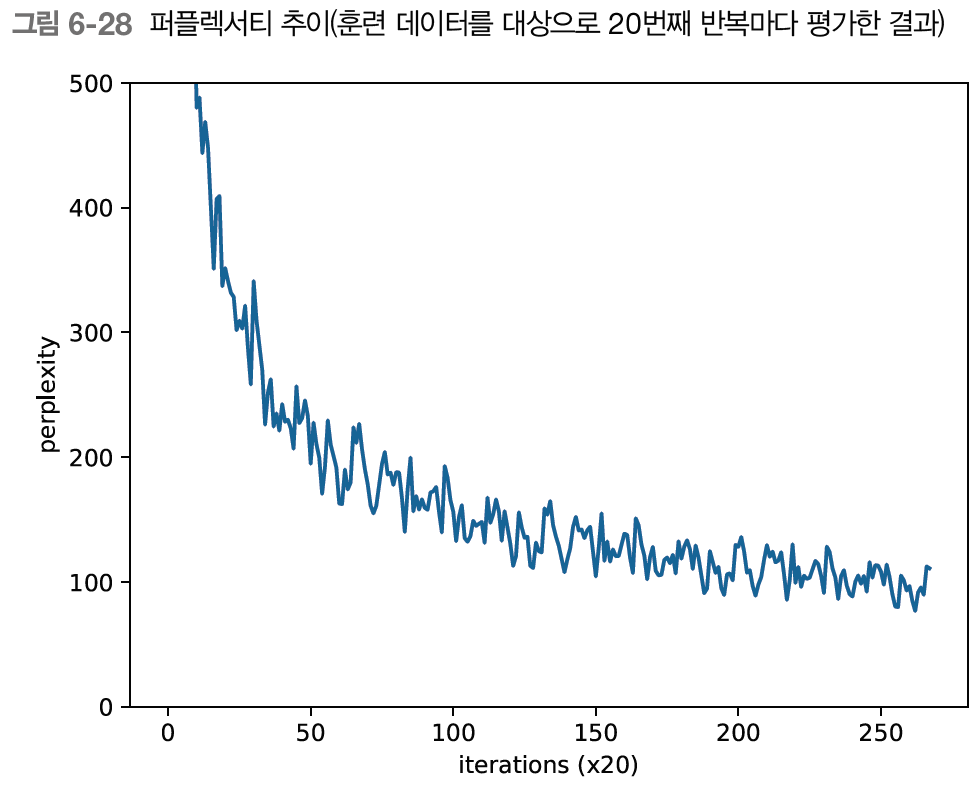
6.4 LSTM을 사용한 언어 모델

5장의 TimeRNN을 TimeLSTM으로 교체



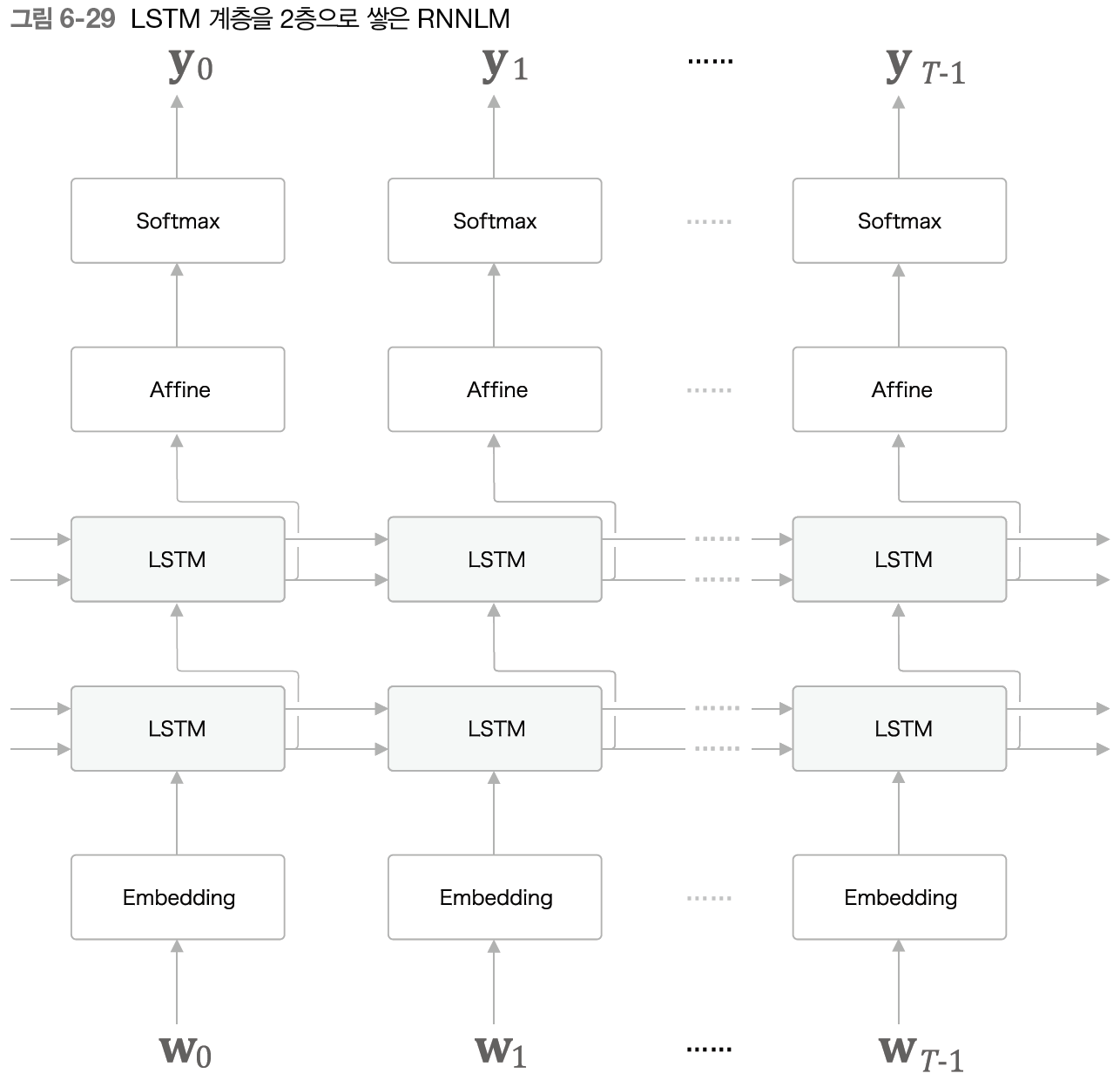
ch06/rnnlm.py 읽기

ch06/train\_rnnlm.py 읽기



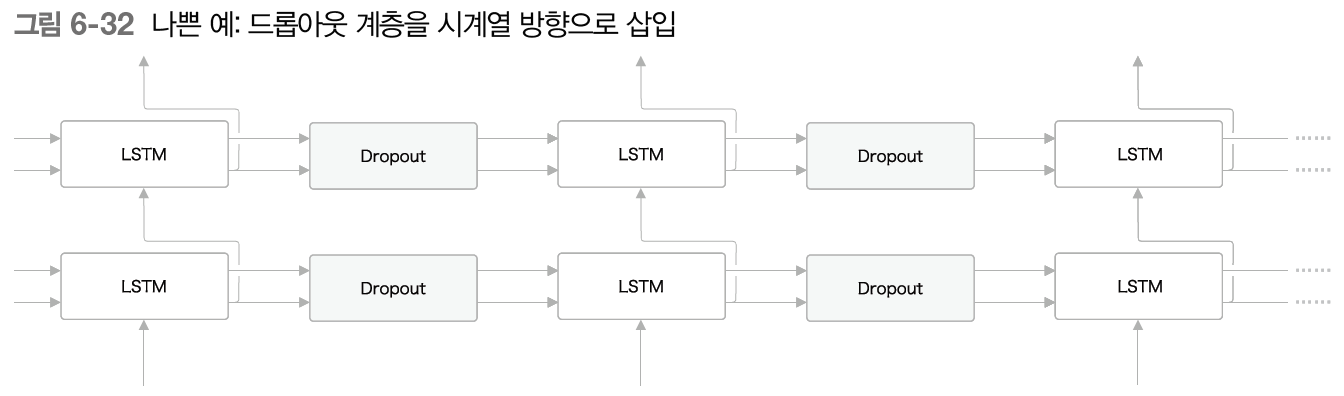
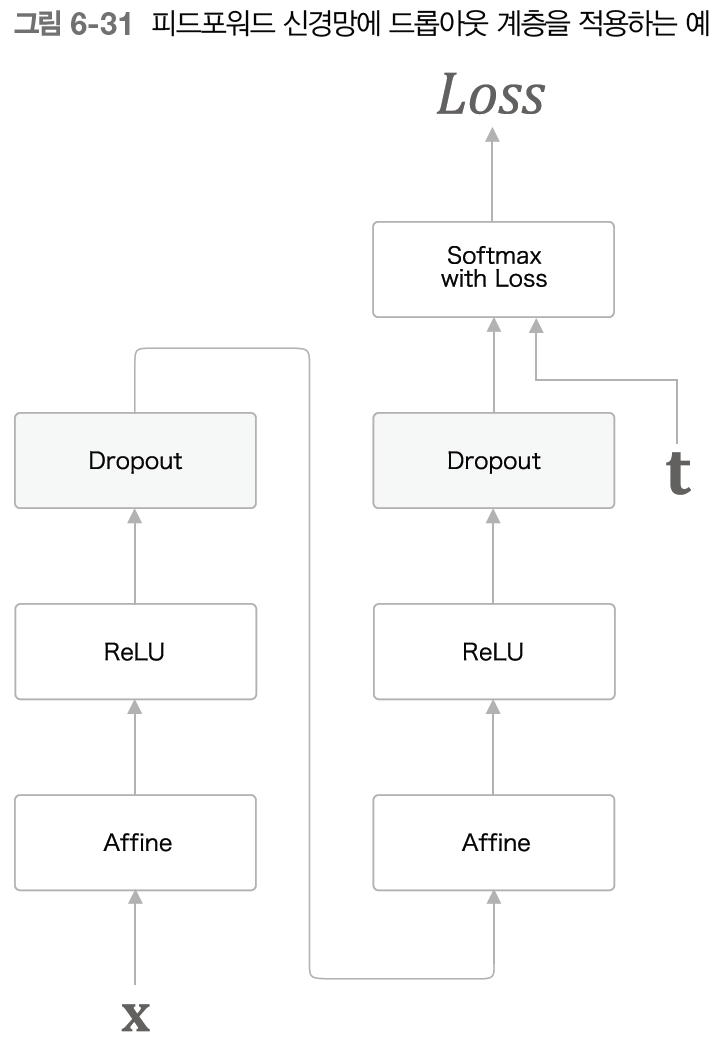
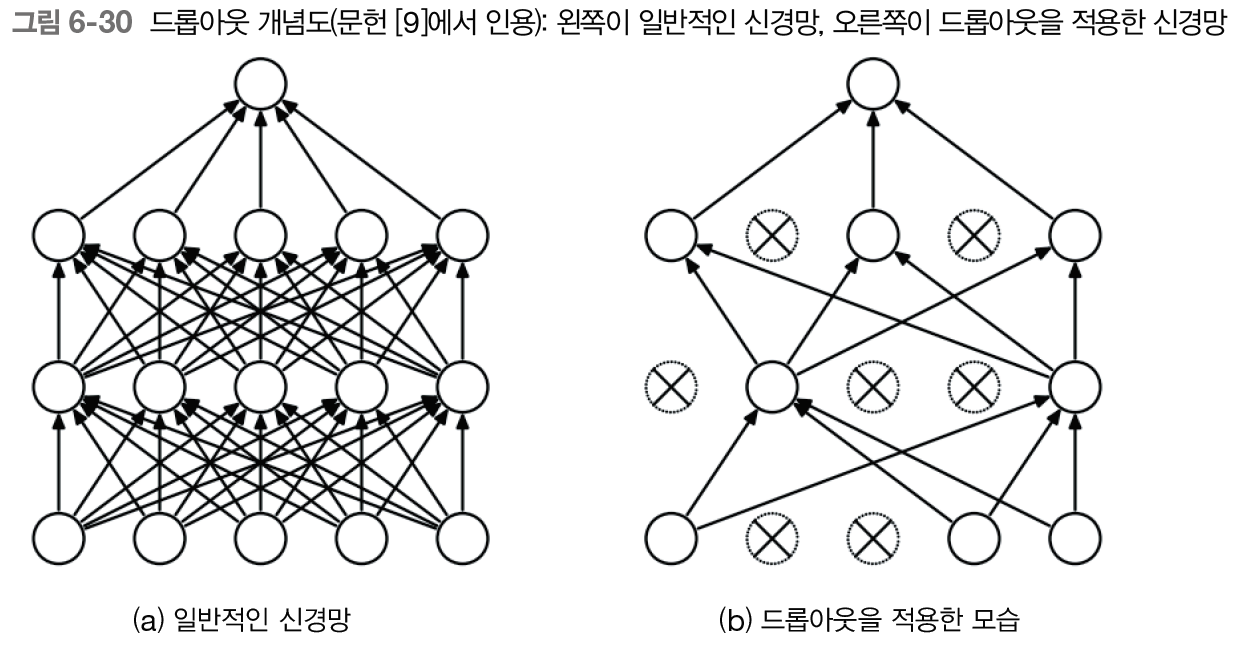
6.5 RNNLM 추가 개선

6.5.1 LSTM 계층 다층화

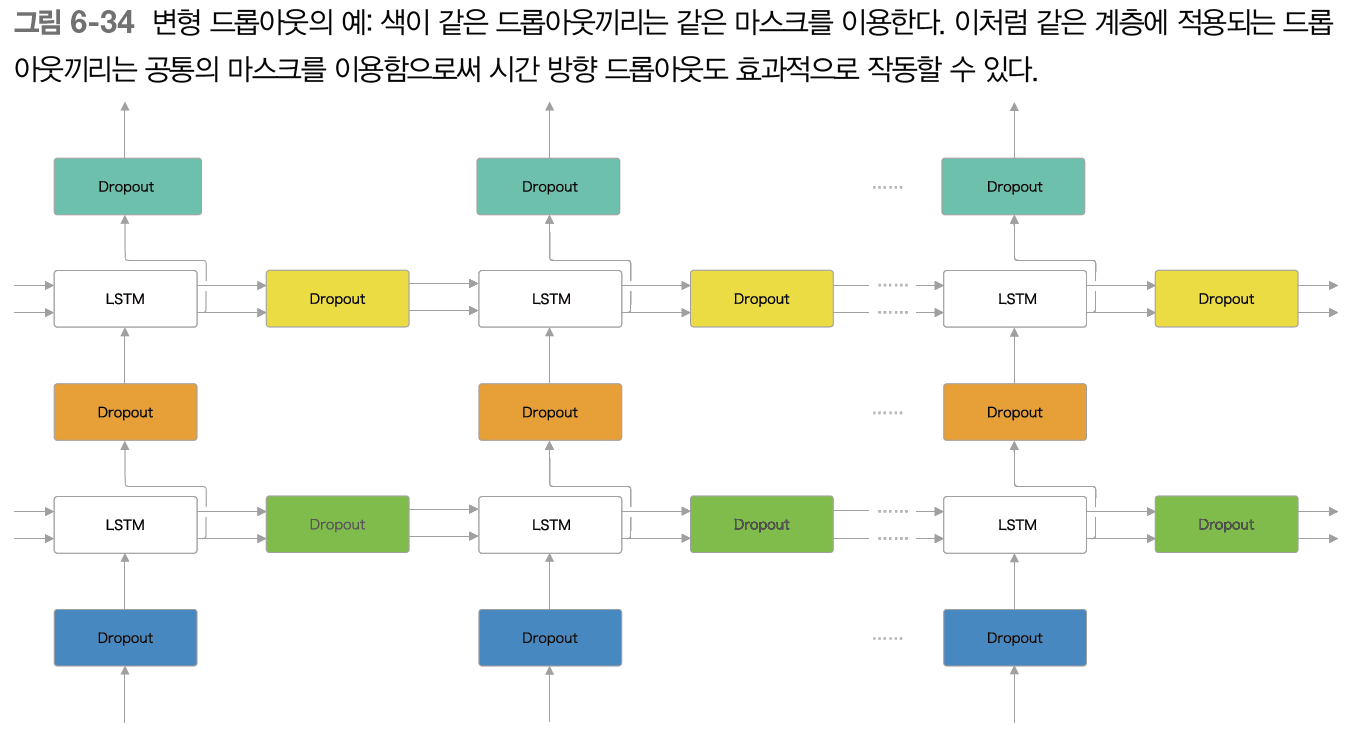
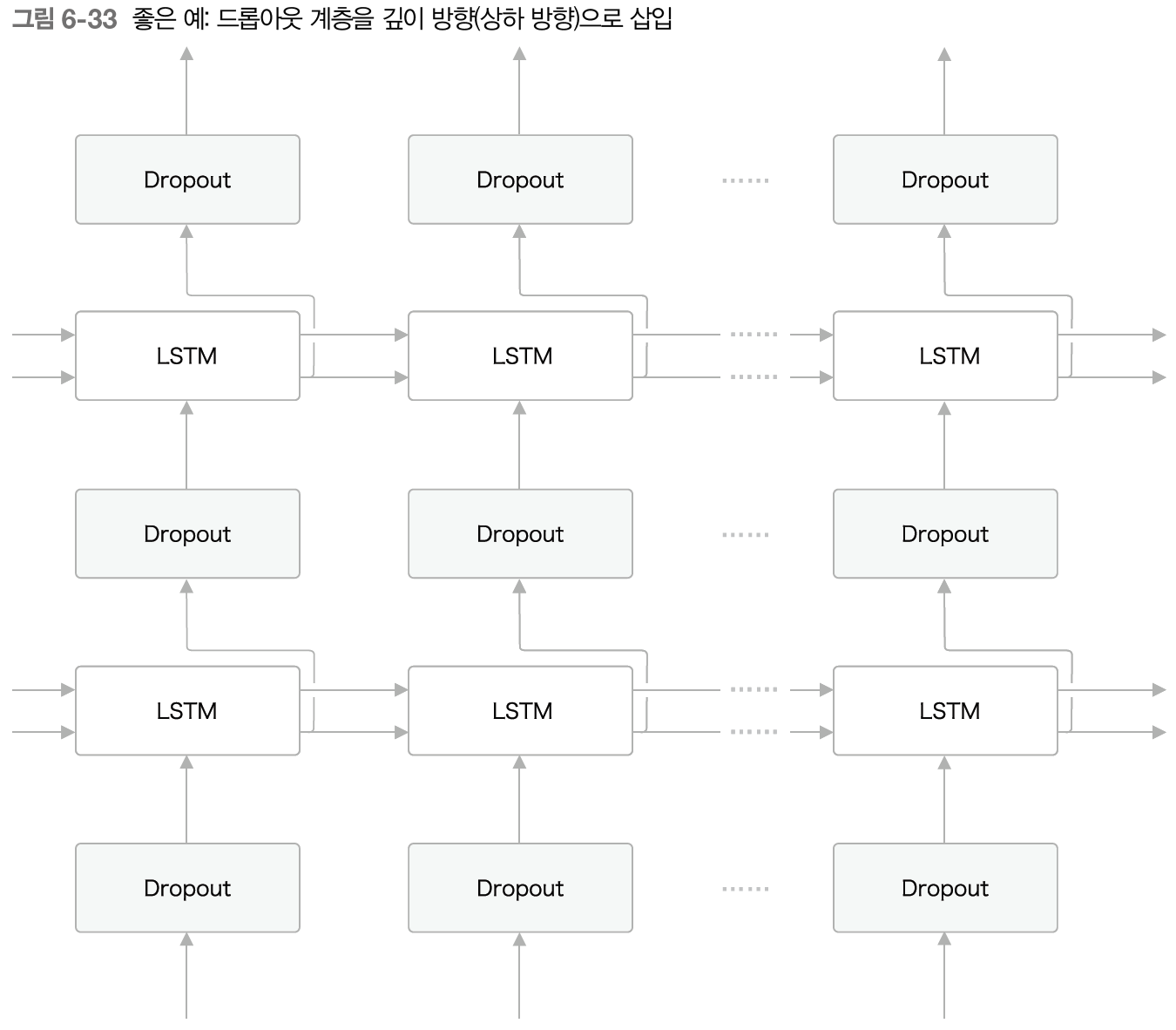


구글 번역에서 사용하는 GNMT는 LSTM 8층

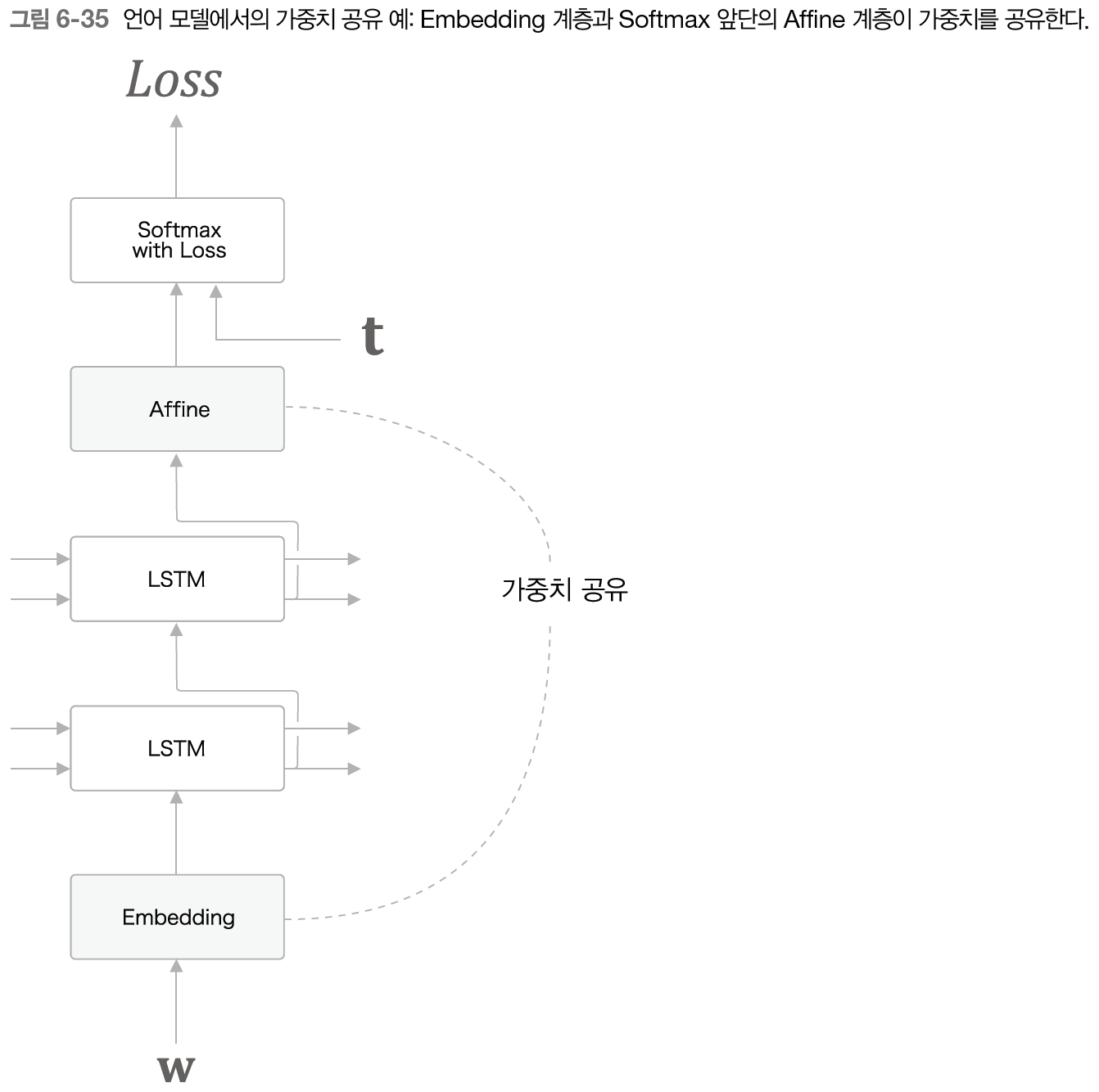
6.5.2 드롭아웃에 의한 과적합 억제



시간 방향 정보 상실

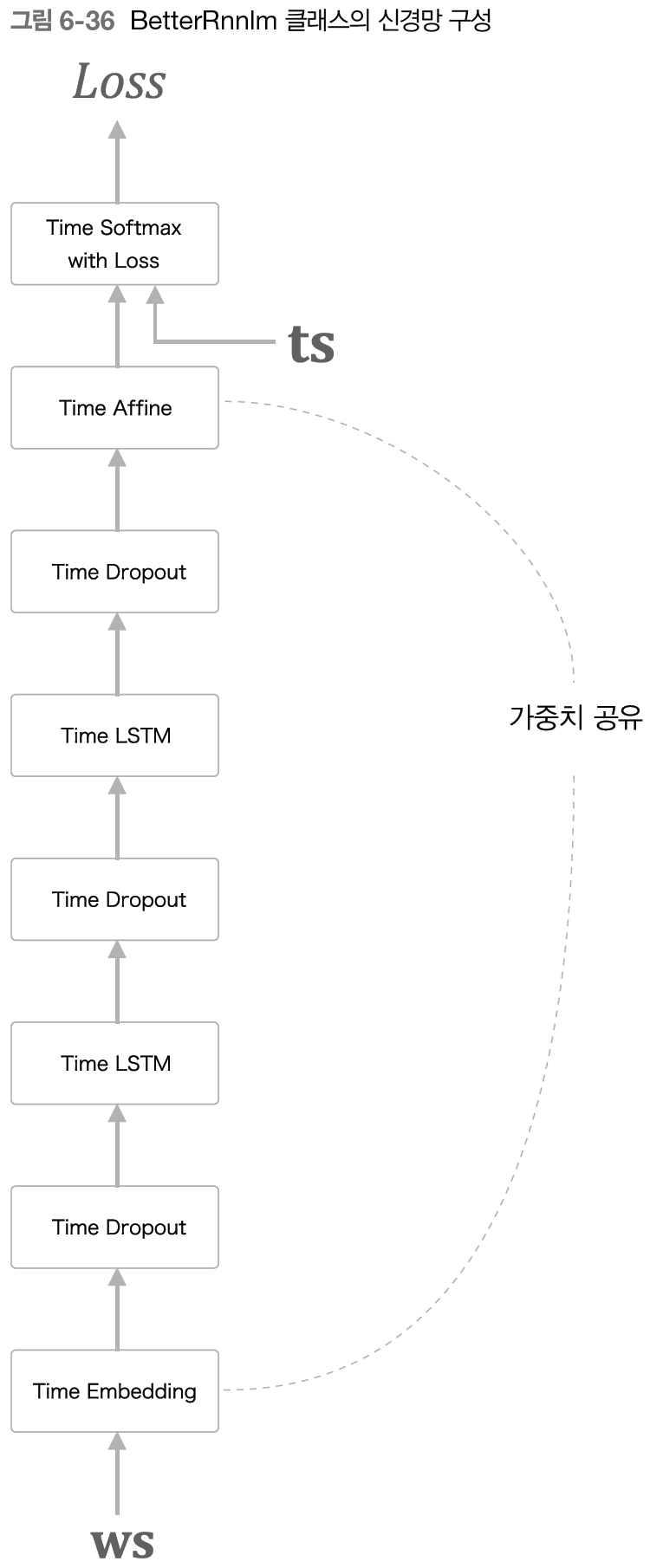


6.5.3 가중치 공유



가중치를 전치해서 V H 차원을 H V 차원으로 사용

6.5.4 개선된 RNNLM 구현



ch06/better\_rnnlm.py 읽기

학습방법 개선

검증데이터의 perplexity가 나빠졌을 경우에만 학습률을 낮춤

ch06/train\_better\_rnnlm.py 읽기