인 공 지 능

[순환 신경망 II]

jaekoo@kookmin.ac.kr

KMU 국민대학교 KOOKMIN UNIVERSITY

8.3 장기 문맥 의존성

장기 문맥 의존성long-term dependency

- 관련된 요소가 멀리 떨어진 상황
- 예, 아래 문장에서 순간 t=1의 '길동은'과 순간 t=32의 '쉬기로'는 아주 밀접한 관련

"길동은, 어제는 친구와 소풍을 다녀왔고, 글피는 엄마를 따라 시장에 가서 반찬거리를 사 오고, 그글피는 여자 친구와 함께 비가 옴에도 불구하고 놀이동산에서 재미있게 놀고 왔기 때문에 오늘은 집에서 푹 쉬기로 작정하였다."

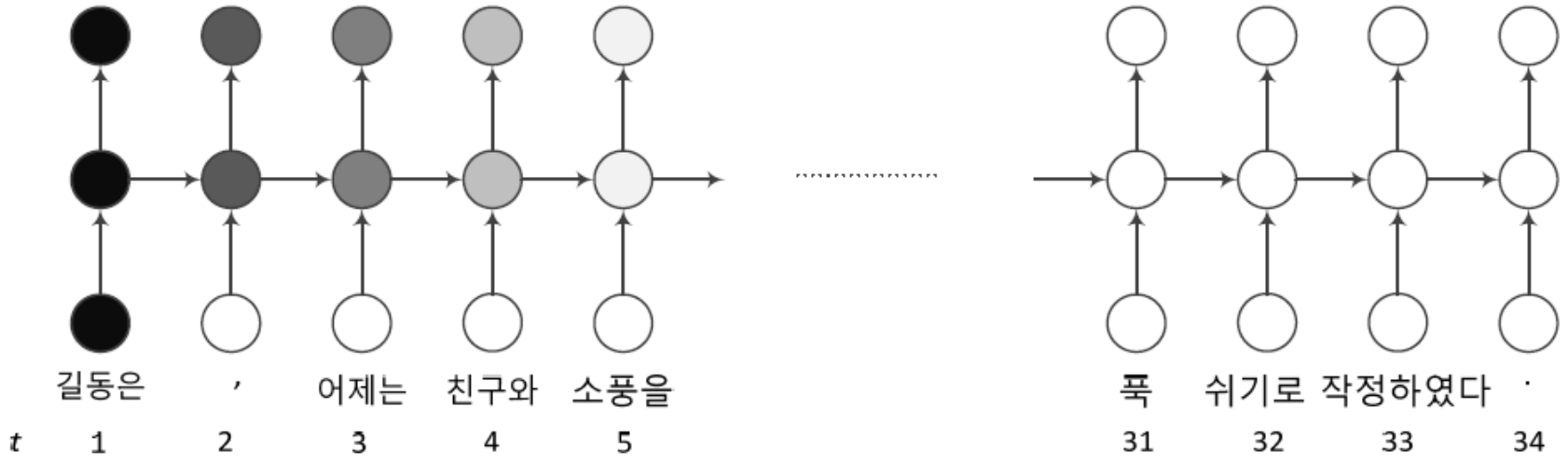


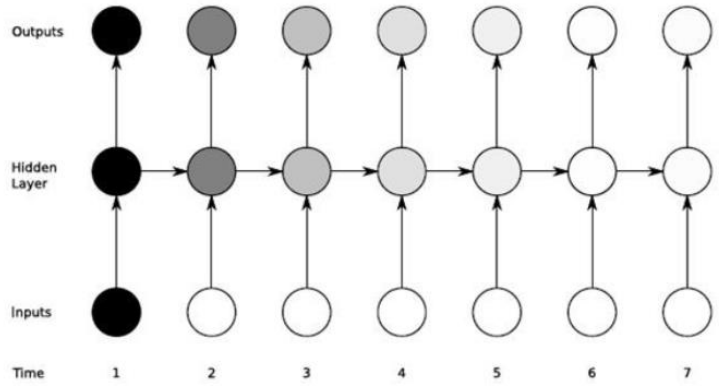
그림 8-13 긴 순차 데이터에서 영향력 감쇠 현상

8.3 장기 문맥 의존성

문제점

- 경사 소멸gradient vanishing (**W** 요소가 1보다 작을 때)

  또는 경사 폭발gradient exploding (**W** 요소가 1보다 클때)



$$\frac{\partial output}{\partial hidden2}\frac{\partial hidden2}{\partial hidden1} = \frac{\partial Sigmoid(z_1)}{\partial z_1}\overset{<1}{w3} * \frac{\partial Sigmoid(z_2)}{\partial z_2}\overset{<1}{w2}$$

$$\overset{<1/4}{\qquad\qquad}\qquad\overset{<1/4}{\qquad\qquad}$$

$$a_h^t = \sum_{i=1}^{I} w_{ih} x_i^t + \sum_{h'=1}^{H} w_{h'h} b_{h'}^{t-1}$$

$$b_h^t = \theta_h(a_h^t)$$

- RNN은 DMLP나 CNN보다 심각
  - 긴 입력 샘플이 자주 발생하기 때문
  - 가중치 공유 때문에 같은 값을 계속 곱함

LSTM은 가장 널리 사용되는 해결책

8.4 LSTM(long short term memory)

- 8.4.1 개폐구gate를 이용한 영향력 범위 확장
- 8.4.2 LSTM의 동작
- 8.4.3 망각 개폐구와 작은 구멍pinhole

8.4.1 개폐구gate를 이용한 영향력 범위 확장

입력 개폐구와 출력 개폐구

- 개폐구를 열면( ) 신호가 흐르고, 닫으면( ) 차단됨
- 예, [그림 8-14]에서 t=1에서는 입력만 열렸고, 32와 33에서는 입력과 출력이 모두 열림
- 실제로는 [0,1] 사이의 실수 값으로 개폐 정도를 조절 ← 이 값은 학습으로 알아냄

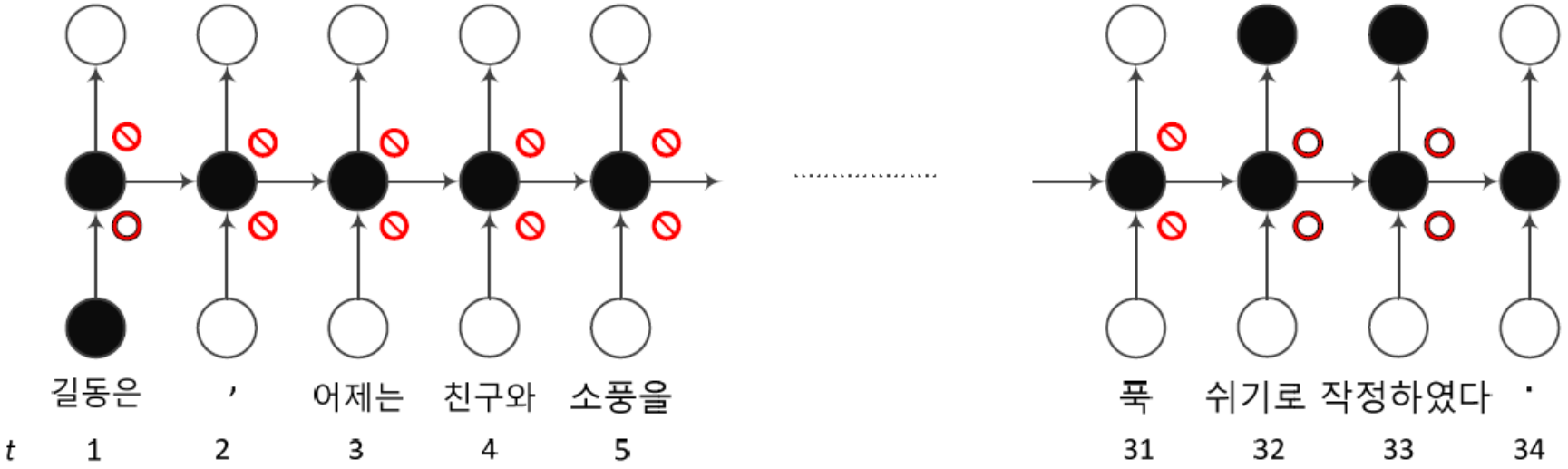
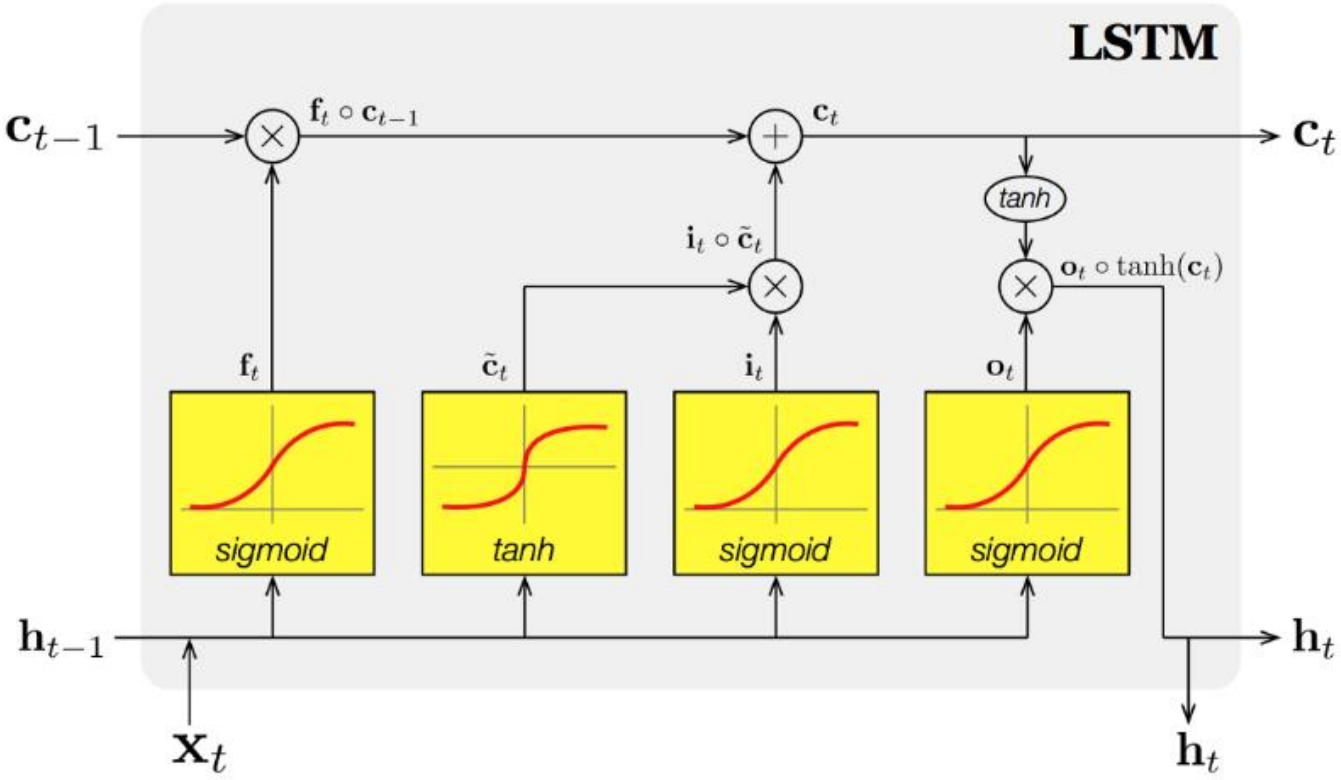


그림 8-14 입력 게이트와 출력 게이트를 이용한 입출력 제어

8.4.1 개폐구gate를 이용한 영향력 범위 확장

LSTM 핵심 요소

- 메모리 블록 (셀): 은닉 상태hidden state 장기 기억
- 망각forget 개폐구 (1: 유지, 0: 제거): 기억 유지 혹은 제거
- 입력input 개폐구: 입력 연산
- 출력output 개폐구: 출력 연산



Gating variables

$$\mathbf{f}_t = \sigma\left(\mathbf{W}_f[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_t\right)$$
$$\mathbf{i}_t = \sigma\left(\mathbf{W}_i[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i\right)$$
$$\mathbf{o}_t = \sigma\left(\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o\right)$$

Candidate (memory) cell state

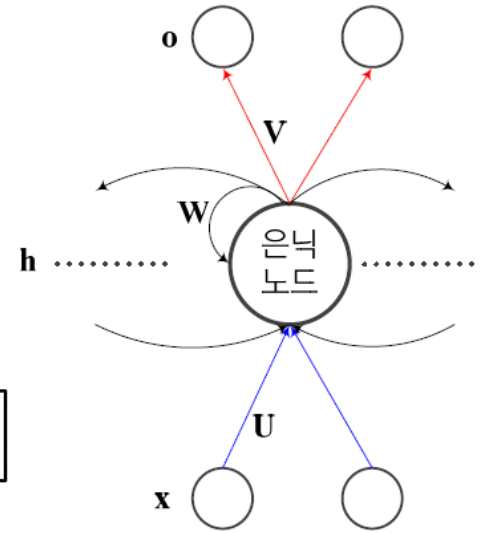
$$\tilde{\mathbf{c}}_t = \tanh\left(\mathbf{W}_c[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_c\right)$$

Cell & Hidden state

$$\mathbf{c}_t = \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \tilde{\mathbf{c}}_t$$
$$\mathbf{h}_t = \mathbf{o}_t \circ \tanh(\mathbf{c}_t)$$

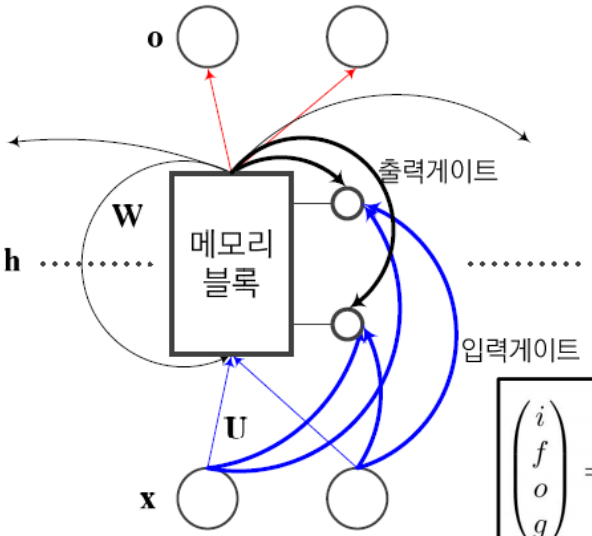
8.4.1 개폐구gate를 이용한 영향력 범위 확장

- RNN과 LSTM의 비교
  - [그림 8-15(a)]의 RNN은 [그림 8-4(a)]를 다른 형태로 그린 것 (LSTM과 비교 목적)
    - 정보 흐름 (얇은 선): 입력→은닉 [파랑], 은닉→은닉 [검정], 은닉→출력 [빨강]



(a) RNN의 은닉 노드

$$h_t = \tanh\left(W\begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right)$$



(b) LSTM의 은닉 노드

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

$$c_t = f \odot c_{t-1} + i \odot g$$

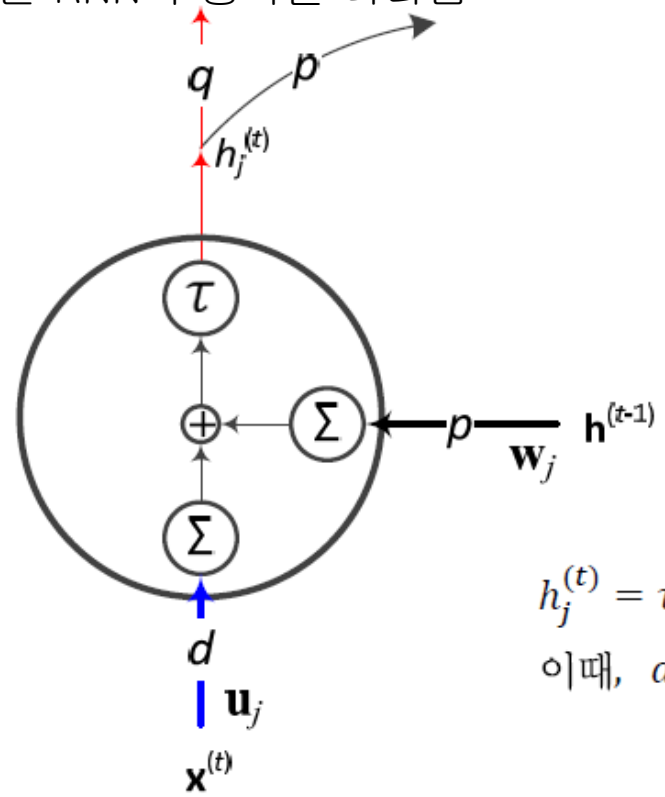
$$h_t = o \odot \tanh(c_t)$$

그림 8-15 RNN과 LSTM의 비교

  - [그림 8-15(b)]의 LSTM은 메모리 블록을 가짐
    - 정보 흐름 (얇은 선): 입력→은닉 [파랑], 은닉→은닉 [검정], 은닉→출력 [빨강] ← RNN과 동일
    - 추가로,
      - 메모리 블록 (혹은 셀cell)의 출력→출력 개폐구, 입력 개폐구 [굵은 검정]
      - 입력 벡터→출력 개폐구, 입력 개폐구 [굵은 파랑]

8.4.2 LSTM의 동작

- RNN의 은닉 노드를 확대하여 다시 살펴보면,
  - [그림 8-16]은 LSTM과 같은 표기법을 쓰기 위해 다시 그린 것
  - 굵은 선은 가중치 벡터
  - 식 (8.6)은 RNN의 동작을 나타냄



$$h_j^{(t)} = \tau\left(a_j^{(t)}\right), \qquad j = 1,2,\cdots,p$$
$$\text{이때, } a_j^{(t)} = \mathbf{w}_j\mathbf{h}^{(t-1)} + \mathbf{u}_j\mathbf{x}^{(t)} + b_j \tag{8.6}$$

그림 8-16 RNN 은닉 노드의 구조와 동작 다시 보기(j번째 은닉 노드)

8.4.2 LSTM의 동작

LSTM의 동작

- [그림 8-17]의 LSTM에서

  출력 게이트와 입력 개폐구의 값이

  1.0으로 고정되면 RNN 동작과 동일함

- 하지만 이들 값은 가중치와 신호 값에

  따라 정해지며 개폐 정도를 조절함

  ← RNN과 차별성

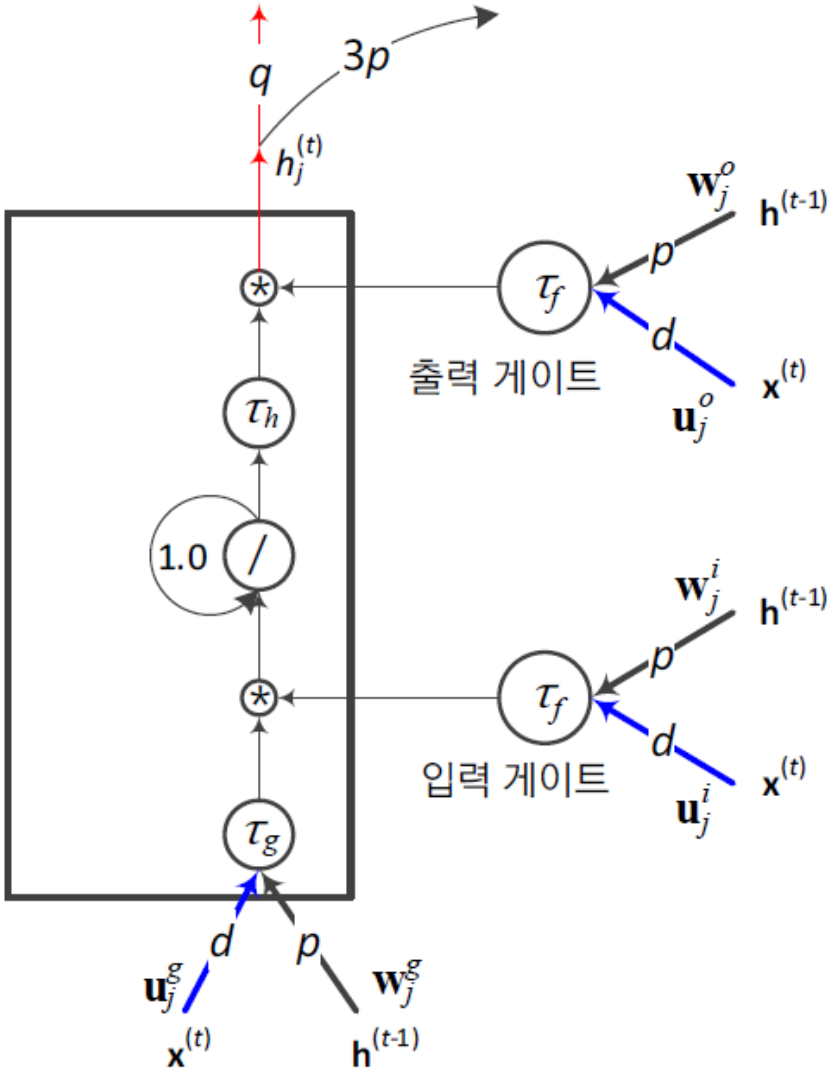


그림 8-17 LSTM 메모리 블록의 구조와 동작(j번째 메모리 블록)

8.4.2 LSTM의 동작

LSTM의 가중치

- 은닉층과 은닉층을 잇는 순환 연결은 세 종류의 가중치를 가짐 (굵은 검정 선)
  - 입력단과 연결하는 $\mathbf{W}^g$, 입력 개폐구와 연결하는 $\mathbf{W}^i$, 출력 개폐구와 연결하는 $\mathbf{W}^o$
- 입력층과 은닉층을 연결하는 가중치 $\mathbf{U}$도 마찬가지(굵은 파란 선)

가중치를 행렬로 표현하면,

$$\mathbf{W}^g = \begin{pmatrix} w_{11}^g & w_{12}^g & \cdots & w_{1p}^g \\ w_{21}^g & w_{22}^g & \cdots & w_{2p}^g \\ \vdots & \vdots & \ddots & \vdots \\ w_{p1}^g & w_{p2}^g & \cdots & w_{pp}^g \end{pmatrix}, \mathbf{W}^i = \begin{pmatrix} w_{11}^i & w_{12}^i & \cdots & w_{1p}^i \\ w_{21}^i & w_{22}^i & \cdots & w_{2p}^i \\ \vdots & \vdots & \ddots & \vdots \\ w_{p1}^i & w_{p2}^i & \cdots & w_{pp}^i \end{pmatrix}, \mathbf{W}^o = \begin{pmatrix} w_{11}^o & w_{12}^o & \cdots & w_{1p}^o \\ w_{21}^o & w_{22}^o & \cdots & w_{2p}^o \\ \vdots & \vdots & \ddots & \vdots \\ w_{p1}^o & w_{p2}^o & \cdots & w_{pp}^o \end{pmatrix} \quad (8.27)$$

$$\mathbf{U}^g = \begin{pmatrix} u_{11}^g & u_{12}^g & \cdots & u_{1d}^g \\ u_{21}^g & u_{22}^g & \cdots & u_{2d}^g \\ \vdots & \vdots & \ddots & \vdots \\ u_{p1}^g & u_{p2}^g & \cdots & u_{pd}^g \end{pmatrix}, \mathbf{U}^i = \begin{pmatrix} u_{11}^i & u_{12}^i & \cdots & u_{1d}^i \\ u_{21}^i & u_{22}^i & \cdots & u_{2d}^i \\ \vdots & \vdots & \ddots & \vdots \\ u_{p1}^i & u_{p2}^i & \cdots & u_{pd}^i \end{pmatrix}, \mathbf{U}^o = \begin{pmatrix} u_{11}^o & u_{12}^o & \cdots & u_{1d}^o \\ u_{21}^o & u_{22}^o & \cdots & u_{2d}^o \\ \vdots & \vdots & \ddots & \vdots \\ u_{p1}^o & u_{p2}^o & \cdots & u_{pd}^o \end{pmatrix} \quad (8.28)$$

8.4.2 LSTM의 동작

- 세 곳 (입력단, 입력 개폐구input gate, 출력 개폐구output gate)에서의 계산

$$\text{입력단: } g = \tau_g\left(\mathbf{u}_j^g \mathbf{x}^{(t)} + \mathbf{w}_j^g \mathbf{h}^{(t-1)} + b_j^g\right) \quad (8.29)$$

$$\text{입력 게이트: } i = \tau_f\left(\mathbf{u}_j^i \mathbf{x}^{(t)} + \mathbf{w}_j^i \mathbf{h}^{(t-1)} + b_j^i\right) \quad (8.30)$$

$$\text{출력 게이트: } o = \tau_f\left(\mathbf{u}_j^o \mathbf{x}^{(t)} + \mathbf{w}_j^o \mathbf{h}^{(t-1)} + b_j^o\right) \quad (8.31)$$

  - g, i, o 값은 가중치 $\mathbf{u}, \mathbf{w}$, 현재 순간의 입력벡터 $\mathbf{x}^{(t)}$, 이전 순간의 상태 $\mathbf{h}^{(t-1)}$에 따라 결정됨

    → 이들 값에 따라 개폐 정도가 정해짐

  - τ_g는 tanh, τ_f는 logistic sigmoid 주로 사용

- 아래쪽 곱 기호 *는 개폐를 조절하는 역할
  - 입력 개폐구의 값 i가 0.0에 가깝다면 $g * i$는 0.0에 가깝게 되어 입력단을 차단, 1.0에 가깝다면 그대로 전달하는 효과

8.4.2 LSTM의 동작

- 기호 /가 붙어 있는 원은 메모리 블록의 상태
  - 메모리 블록이 기억하는 내용으로 시간에 따라 변하므로 $s^{(t)}$로 표기

$$s^{(t)} = s^{(t-1)} + g * i \qquad (8.32)$$

  - 해석해 보면,
    - 입력 개폐구의 값 (i)이 0.0이면 $g * i$는 0이 되어 이전 상태와 같게 됨
      (입력 개폐구가 차단되어 이전 내용을 그대로 기억)
      → 이전 입력의 영향력을 더 멀리 확장하는 효과

- 위쪽 곱 기호 *는 개폐를 조절하는 역할
  - 출력 개폐구의 값 (o)이 개폐 정도를 조절

$$h_j^{(t)} = \tau_h(s^{(t)}) * o \qquad (8.33)$$

- 식 (8.33)의 계산 결과인 $h_j^{(t)}$는
  - Q개의 출력 노드로 전달되어 출력단 계산에 사용 (즉 식 (8.8)의 벡터 $\mathbf{h}^{(t)}$의 j번째 요소임)
  - 입력단, 입력 개폐구, 출력 개폐구에 있는 노드로 전달되어 t+1 순간의 계산에 이용됨

8.4.2 LSTM의 동작

- 지금까지 수식을 정리하면,

입력단: $$\mathbf{g} = \tau_g\left(\mathbf{U}^g\mathbf{x}^{(t)} + \mathbf{W}^g\mathbf{h}^{(t-1)} + \mathbf{b}^g\right) \quad (8.34)$$

입력 게이트: $$\mathbf{i} = \tau_f\left(\mathbf{U}^i\mathbf{x}^{(t)} + \mathbf{W}^i\mathbf{h}^{(t-1)} + \mathbf{b}^i\right) \quad (8.35)$$

출력 게이트: $$\mathbf{o} = \tau_f\left(\mathbf{U}^o\mathbf{x}^{(t)} + \mathbf{W}^o\mathbf{h}^{(t-1)} + \mathbf{b}^o\right) \quad (8.36)$$

$$\mathbf{s}^{(t)} = \mathbf{s}^{(t-1)} + \mathbf{g} \odot \mathbf{i} \quad (8.37)$$

$$\mathbf{h}^{(t)} = \tau_h\left(\mathbf{s}^{(t)}\right) \odot \mathbf{o} \quad (8.38)$$

$$\mathbf{y}'^{(t)} = \text{softmax}(\mathbf{V}\mathbf{h}^{(t)} + \mathbf{c}) \quad (8.39)$$

8.4.3 망각 개폐구와 작은 구멍pinhole

- 망각 개폐구forget gate에 의한 LSTM의 확장
  - 이전 순간의 상태 $h^{(t-1)}$ (즉 메모리 블록의 기억)을 지우는 효과

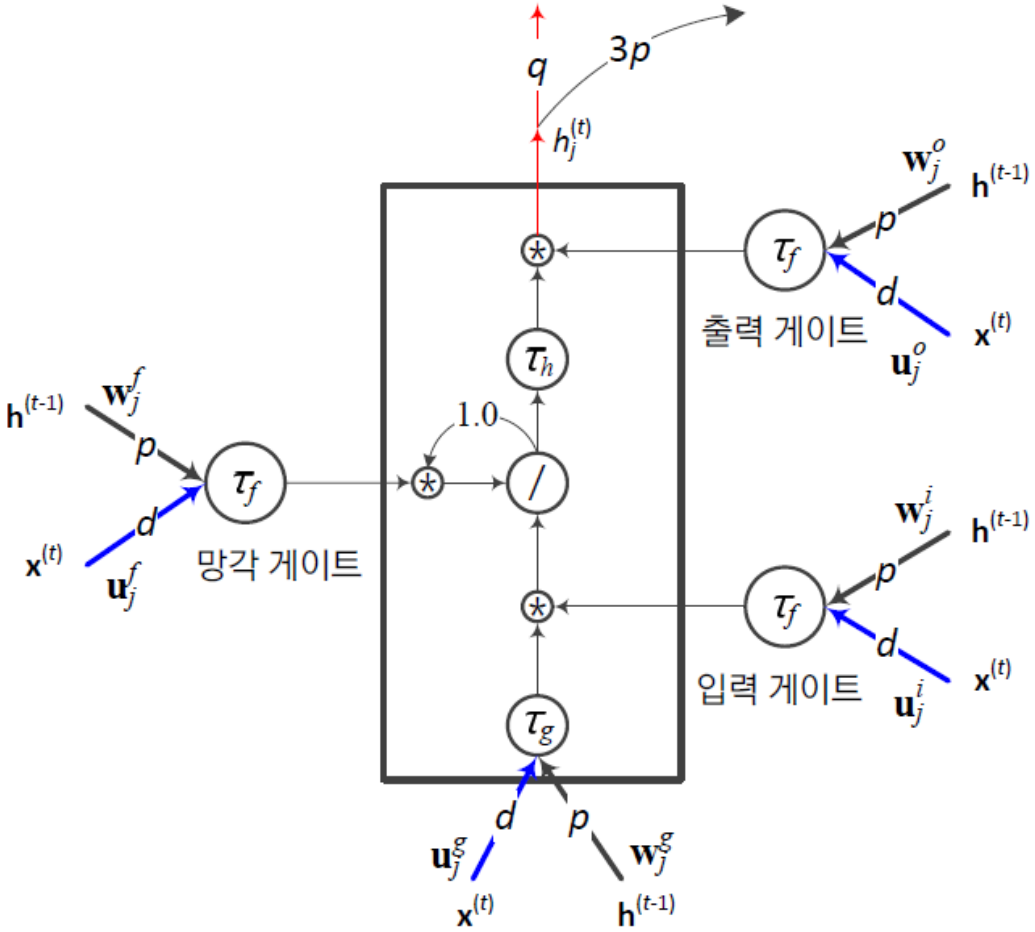


그림 8-18 망각 게이트가 추가된 LSTM 메모리 블록

8.4.3 망각 개폐구와 작은 구멍pinhole

- 망각 개폐구를 가진 LSTM의 동작 (파란 박스는 이전과 다른 점)

입력단: $$\mathbf{g} = \tau_g\left(\mathbf{U}^g\mathbf{x}^{(t)} + \mathbf{W}^g\mathbf{h}^{(t-1)} + \mathbf{b}^g\right) \quad (8.40)$$

입력 게이트: $$\mathbf{i} = \tau_f\left(\mathbf{U}^i\mathbf{x}^{(t)} + \mathbf{W}^i\mathbf{h}^{(t-1)} + \mathbf{b}^i\right) \quad (8.41)$$

출력 게이트: $$\mathbf{o} = \tau_f\left(\mathbf{U}^o\mathbf{x}^{(t)} + \mathbf{W}^o\mathbf{h}^{(t-1)} + \mathbf{b}^o\right) \quad (8.42)$$

망각 게이트: $$\mathbf{f} = \tau_f\left(\mathbf{U}^f\mathbf{x}^{(t)} + \mathbf{W}^f\mathbf{h}^{(t-1)} + \mathbf{b}^f\right) \quad (8.43)$$

$$\mathbf{s}^{(t)} = \mathbf{f}\odot\mathbf{s}^{(t-1)} + \mathbf{g}\odot\mathbf{i} \quad (8.44)$$

$$\mathbf{h}^{(t)} = \tau_h\left(\mathbf{s}^{(t)}\right)\odot\mathbf{o} \quad (8.45)$$

$$\mathbf{y}'^{(t)} = \mathrm{softmax}(\mathbf{h}^{(t)}) \quad (8.46)$$

8.4.3 망각 개폐구와 작은 구멍pinhole

- 작은 구멍pinhole 기능으로 LSTM 확장
  - 작은 구멍 (노란색 선)은 블록의 내부 상태를 3개의 개폐구에 알려주는 역할을 함
  - 순차 데이터를 처리하다가 어떤 조건에 따라 특별한 조치를 취해야 하는 응용에 효과적
    - 예, 음성 인식을 수행하다가 특정 단어가 발견되면 지정된 행위를 수행

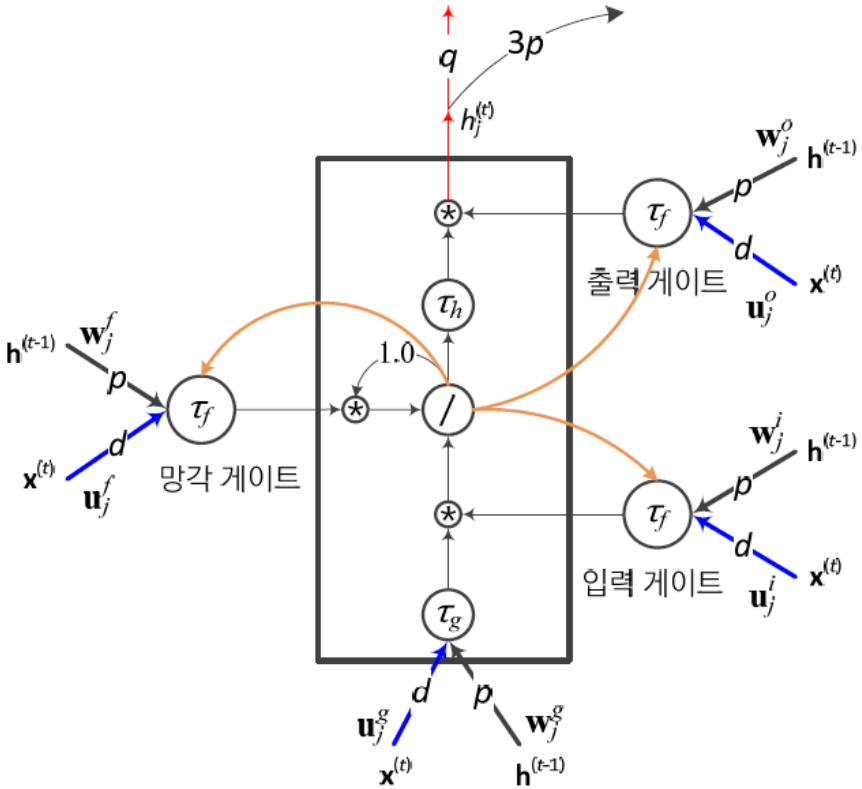
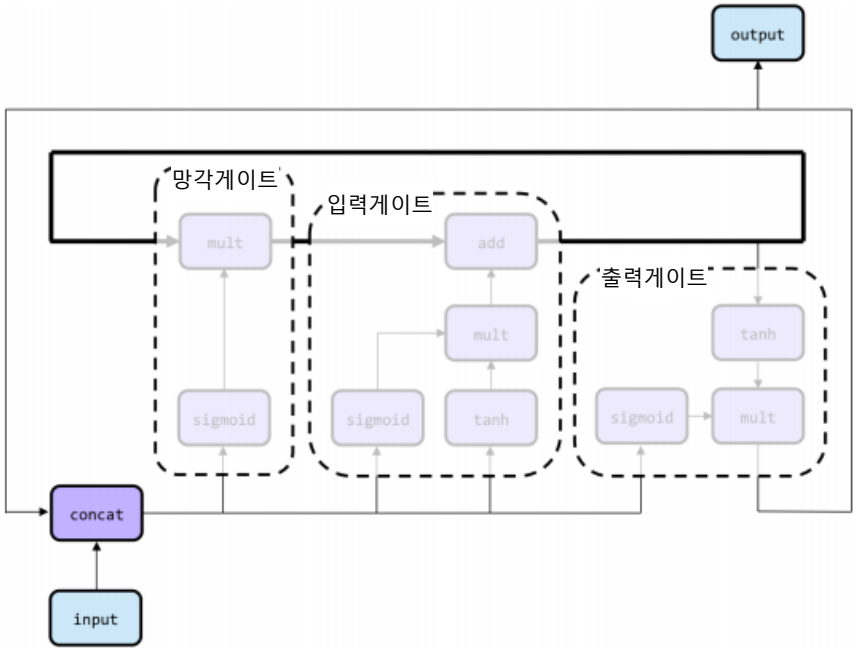


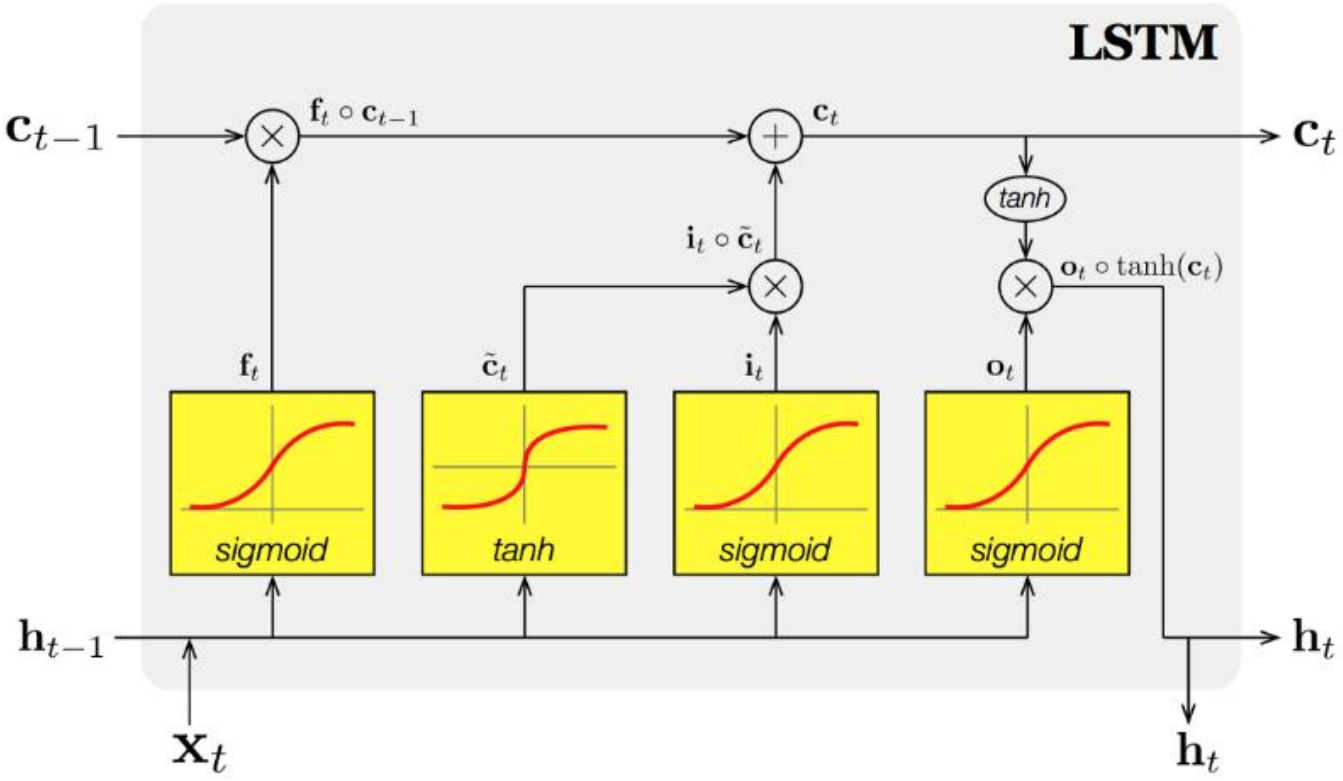
그림 8-19 핍홀이 추가된 LSTM 메모리 블록



LSTM 동작 블록화

8.4.2 LSTM의 동작

- 장기 기억: 메모리 블록 (셀)의 은닉 상태hidden state
- 기억 유지 혹은 제거: 망각 개폐구 (1: 유지, 0: 제거)
- 입력: 입력 개폐구
- 출력: 출력 개폐구



Gating variables

$$\mathbf{f}_t = \sigma\left(\mathbf{W}_f[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_t\right)$$
$$\mathbf{i}_t = \sigma\left(\mathbf{W}_i[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i\right)$$
$$\mathbf{o}_t = \sigma\left(\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o\right)$$

Candidate (memory) cell state

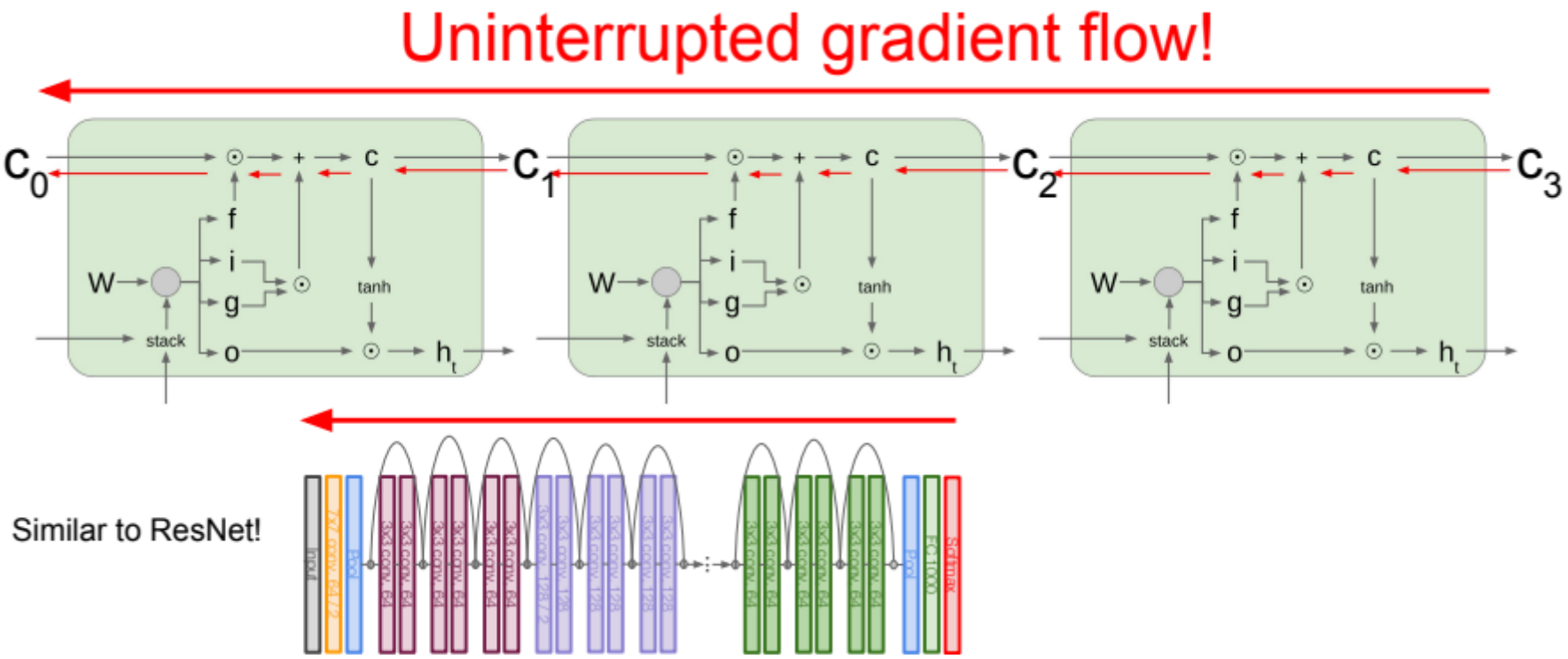
$$\tilde{\mathbf{c}}_t = \tanh\left(\mathbf{W}_c[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_c\right)$$

Cell & Hidden state

$$\mathbf{c}_t = \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \tilde{\mathbf{c}}_t$$
$$\mathbf{h}_t = \mathbf{o}_t \circ \tanh(\mathbf{c}_t)$$

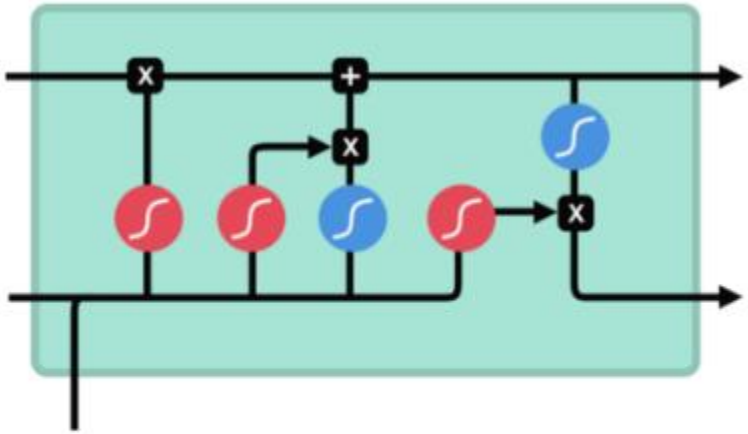
8.4.2 LSTM의 동작

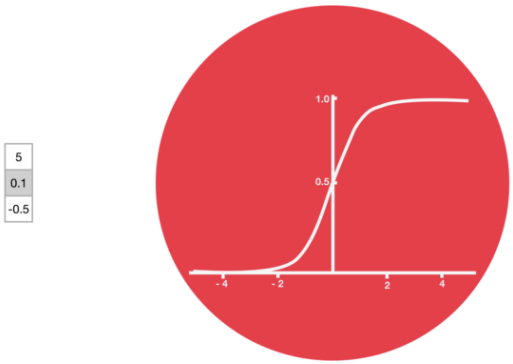
- LSTM 구조에 따른 경사 흐름 개선

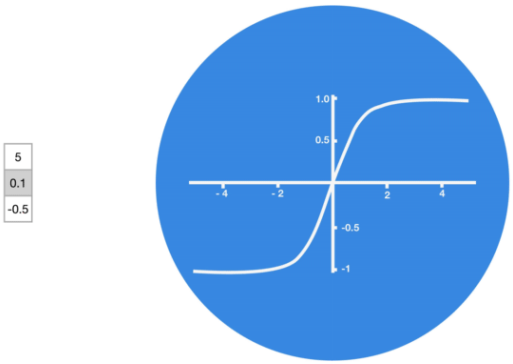


8.4.2 LSTM의 동작

- 전체 동작 예시



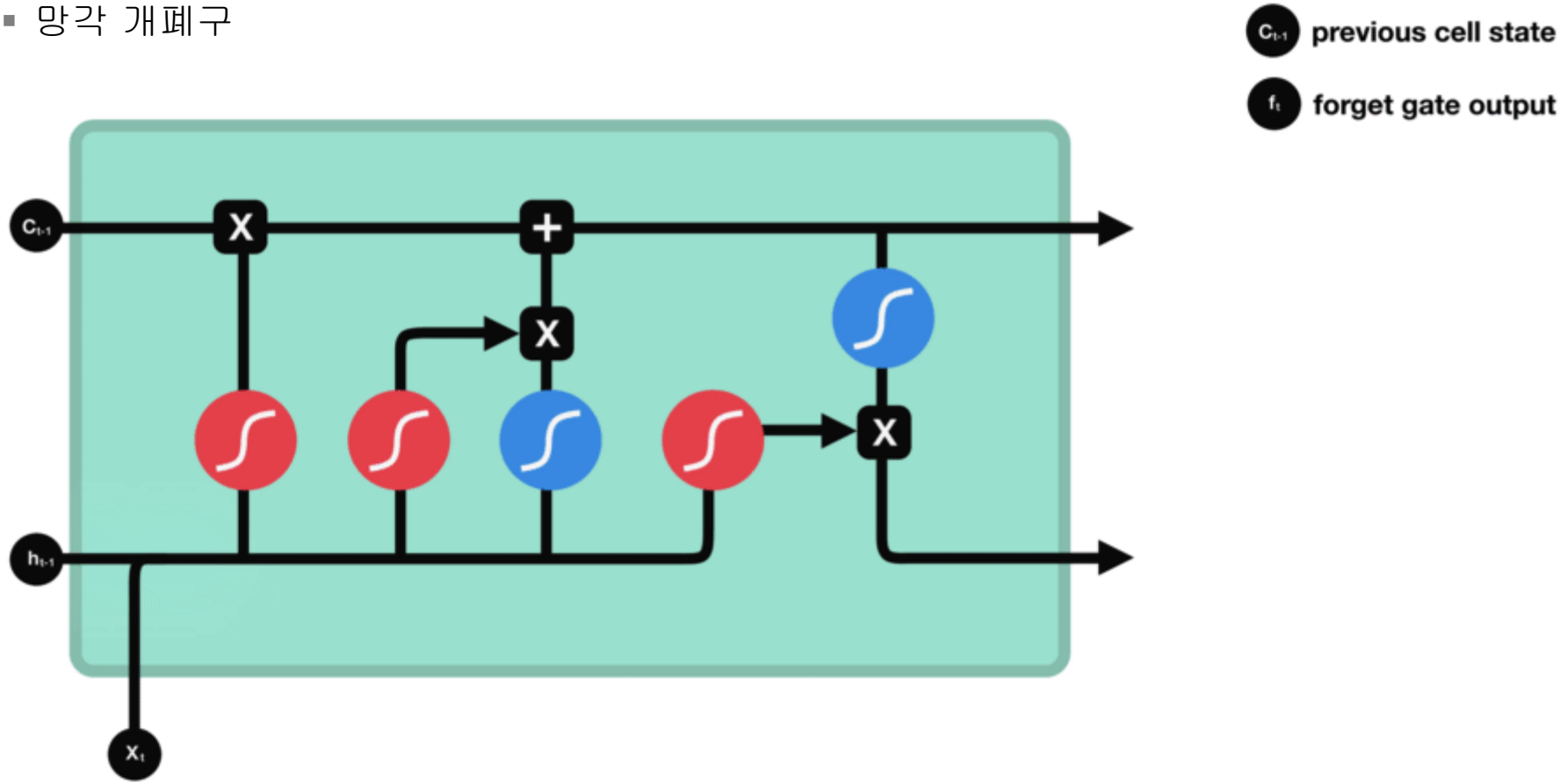




8.4.2 LSTM의 동작

부분 동작 예시

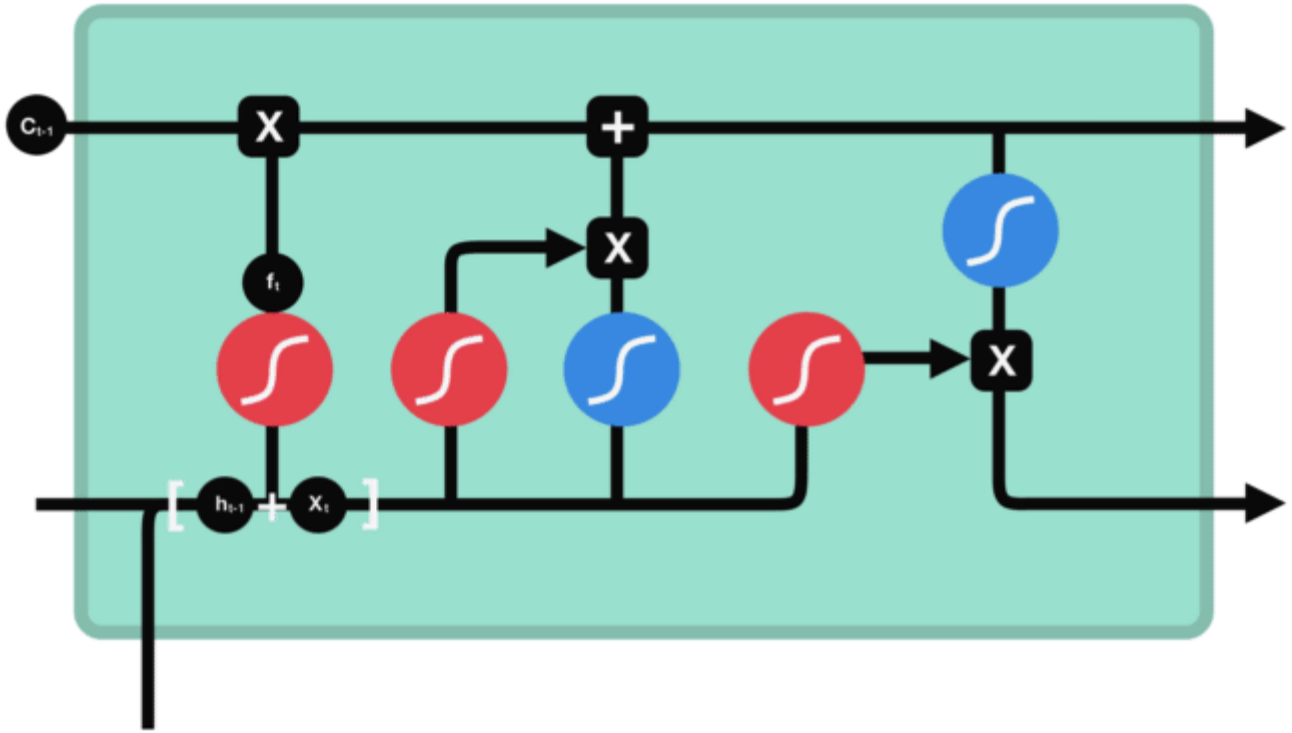
- 망각 개폐구



8.4.2 LSTM의 동작

부분 동작 예시

- 입력 개폐구



C_{t-1} previous cell state

f_t forget gate output

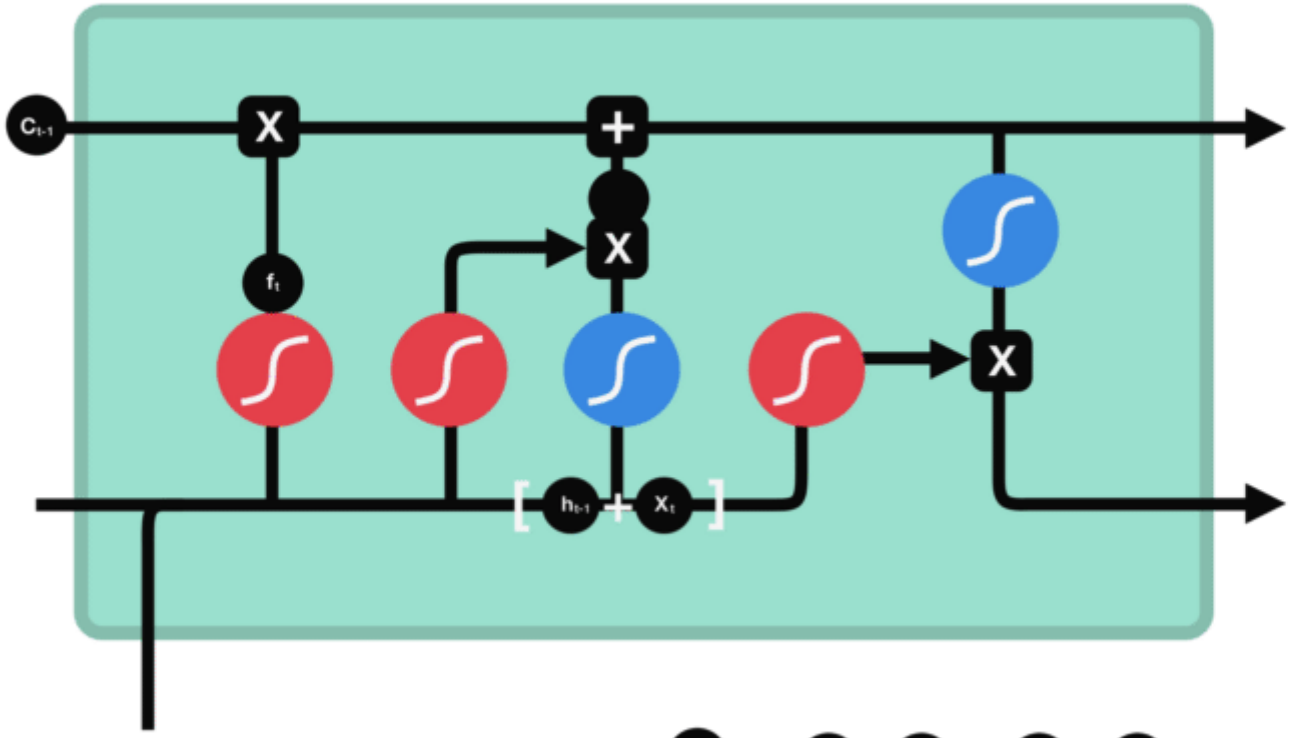
i_t input gate output

$\tilde{C}_t$ candidate

8.4.2 LSTM의 동작

부분 동작 예시

- 셀 상태 갱신



C_{t-1} previous cell state

f_t forget gate output

i_t input gate output

$\tilde{C}_t$ candidate

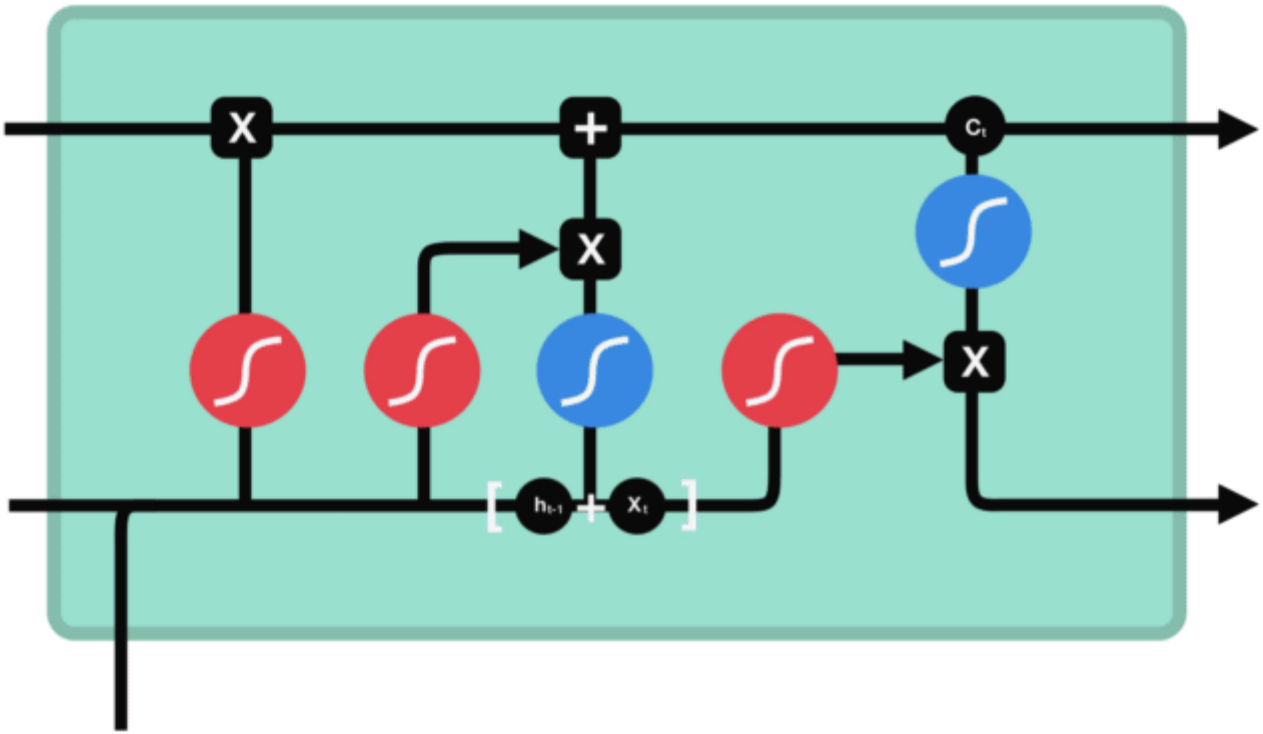
C_t new cell state

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

8.4.2 LSTM의 동작

부분 동작 예시

- 출력 개폐구



c_{t-1} previous cell state

f_t forget gate output

i_t input gate output

$\tilde{c}_t$ candidate

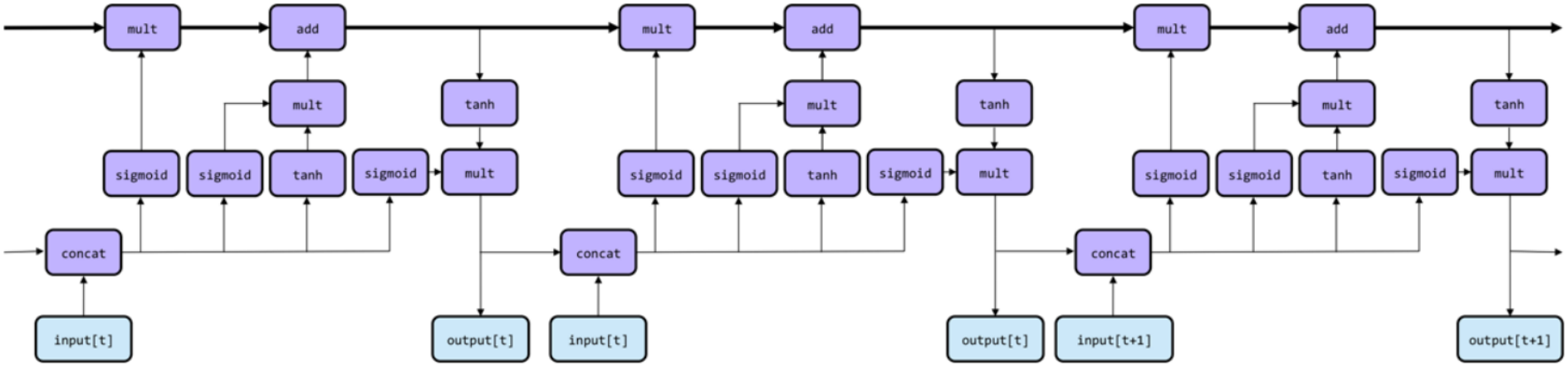
c_t new cell state

o_t output gate output

h_t hidden state

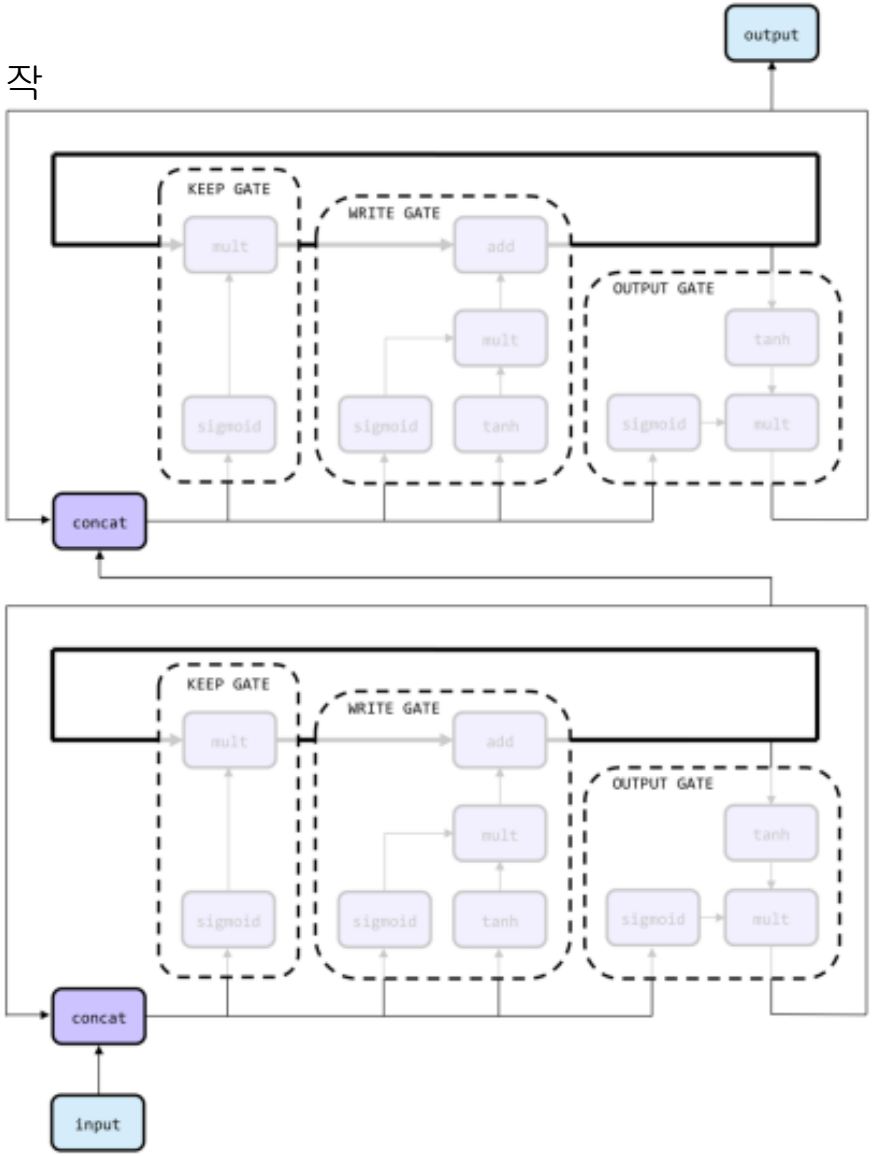
8.4.2 LSTM의 동작

- 시간에 따른 LSTM의 동작



8.4.2 LSTM의 동작

- LSTM 확장
  - stacked LSTM 동작

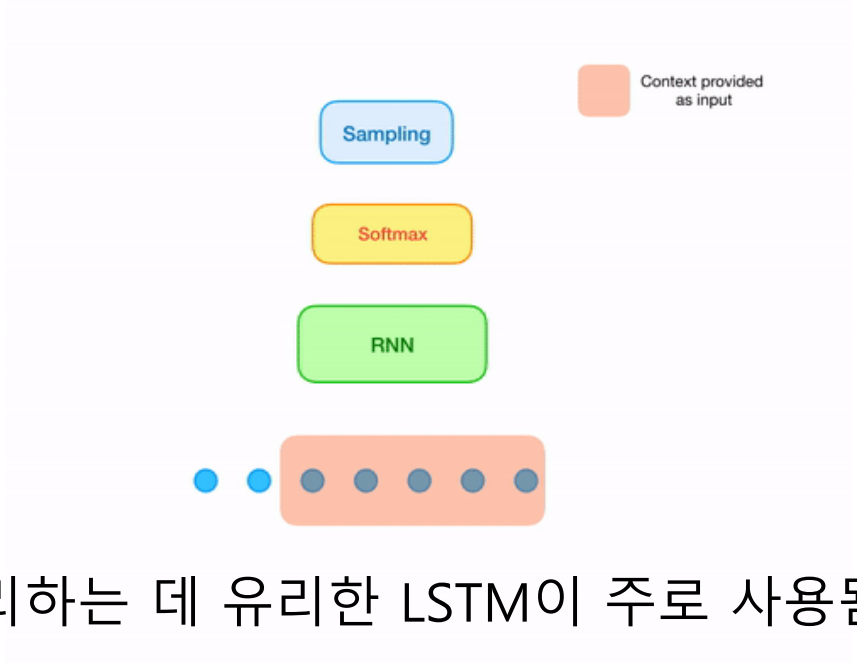


8.5 응용 사례

- 8.5.1 언어 모델
- 8.5.2 기계 번역
- 8.5.3 영상 주석 생성

8.5 응용 사례

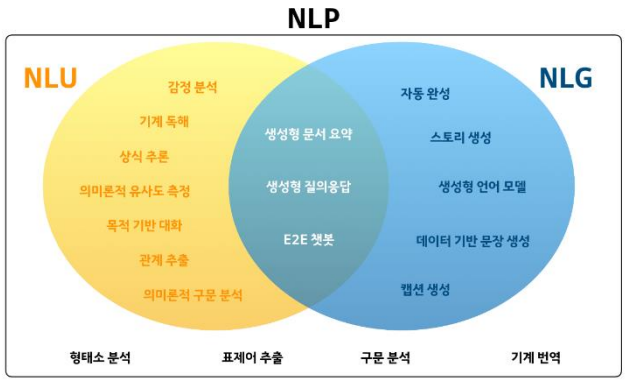
- 순환 신경망은 분별 모델뿐 아니라 생성 모델로도 활용됨



- 장기 문맥을 처리하는 데 유리한 LSTM이 주로 사용됨

8.5.1 언어 모델

언어 모델language model이란

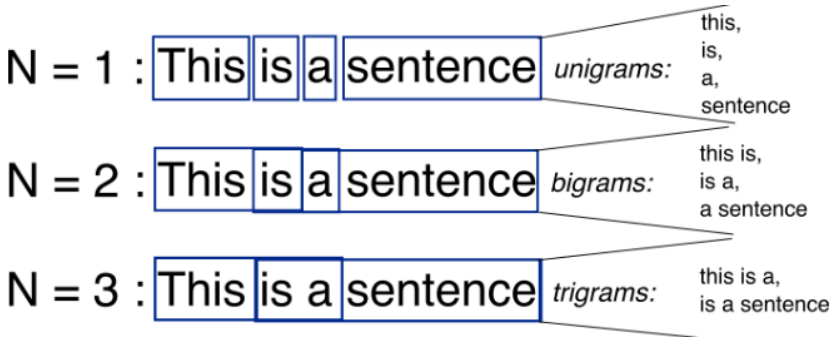


- 문장, 즉 단어 열의 확률분포를 모형화modeling
  - 예, P(자세히, 보아야, 예쁘다) > P(예쁘다, 보아야, 자세히)
- 활용
  - 음성 인식기 또는 언어 번역기가 후보로 출력한 문장이 여럿 있을 때,

    언어 모델로 확률을 계산한 다음 확률이 가장 높은 것을 선택하여 성능을 높임
- 확률분포를 추정하는 방법
  - n-그램gram
  - 다층 퍼셉트론
  - 순환 신경망

8.5.1 언어 모델

n-그램을 이용한 언어 모델



- 고전적인 방법
  - 예로, n=1: unigrams, n=2: bigrams, n=3: trigrams
- 문장을 $\mathbf{x} = (z_1, z_2, \cdots, z_T)^{\mathrm{T}}$라 하면, $\mathbf{x}$가 발생할 확률을 식 (8.47)로 추정

$$P(z_1, z_2, \cdots, z_T) = \prod_{t=1}^{T} P(z_t | z_1, z_2, \cdots, z_{t-1}) \qquad (8.47)$$

- n-그램은 n-1개의 단어만 고려하는데, 이때 식 (8.48)이 성립

$$P(z_1, z_2, \cdots, z_T) \approx \prod_{t=1}^{T} P\left(z_t | z_{t-(n-1)}, \cdots, z_{t-1}\right) \qquad (8.48)$$

  - 알아야 할 확률의 개수는 m^n → 차원의 저주 때문에 n을 1~3 정도로 작게 해야만 함
  - 확률 추정은 말뭉치corpus를 사용
- 단어가 원핫 코드로 표현되므로 단어 간의 의미 있는 거리를 반영하지 못하는 한계

8.5.1 언어 모델

순환 신경망을 이용한 언어 모델

- 현재까지 본 단어 열을 기반으로 다음 단어를 예측하는 방식으로 학습

  → 확률분포 추정뿐만 아니라 문장 생성 기능까지 갖춤

- 비지도 학습에 해당하여 말뭉치로부터 쉽게 훈련집합 구축 가능
- 예, “자세히 보아야 예쁘다”라는 문장은 다음과 같은 샘플이 됨(왼쪽으로 한 칸씩 이동)

$$\mathbf{x} = (< \text{시작} >, \text{자세히}, \text{보아야}, \text{예쁘다})^{\mathrm{T}}, \quad \mathbf{y} = (\text{자세히}, \text{보아야}, \text{예쁘다}, < \text{끝} >)^{\mathrm{T}}$$

- 일반화하면,

$$\mathbf{x} = (< \text{시작} >, \mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_T)^{\mathrm{T}}, \quad \mathbf{y} = (\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_T, < \text{끝} >)^{\mathrm{T}} \tag{8.49}$$

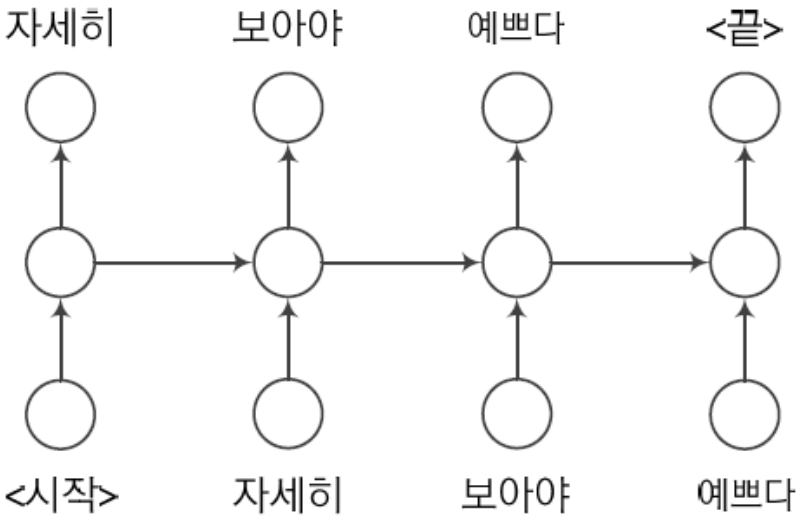


그림 8-20 순환 신경망의 예측 능력을 학습하는 데 사용하는 훈련 샘플

8.5.1 언어 모델

- 순환 신경망의 학습
  - 말뭉치에 있는 문장을 식 (8.49)처럼 변환하여 훈련집합을 만든 다음, BPTT 학습 알고리즘을 적용
- 학습을 마친 순환 신경망 (언어 모델)의 활용
  - 기계 번역기나 음성 인식기의 성능을 향상하는 데 활용
  - 예, 음성 인식기가

    $\tilde{\mathbf{x}}_1 = (\text{자세히}, \text{보아야}, \text{예쁘다})^{\mathrm{T}}$와 $\tilde{\mathbf{x}}_2 = (\text{자세를}, \text{모아야}, \text{예쁘다})^{\mathrm{T}}$라는 2개 후보를 출력했을 때

    언어 모델로 $P(\tilde{\mathbf{x}}_i)$를 계산한 후, 높은 확률의 후보를 선택
- 일반적으로 사전학습을 수행한 언어 모델을 개별 과제에 맞게 미세 조정함

8.5.1 언어 모델

생성 모델로 활용

- 문장 생성한 예 [Karpathy2015]
- 문장 생성 알고리즘

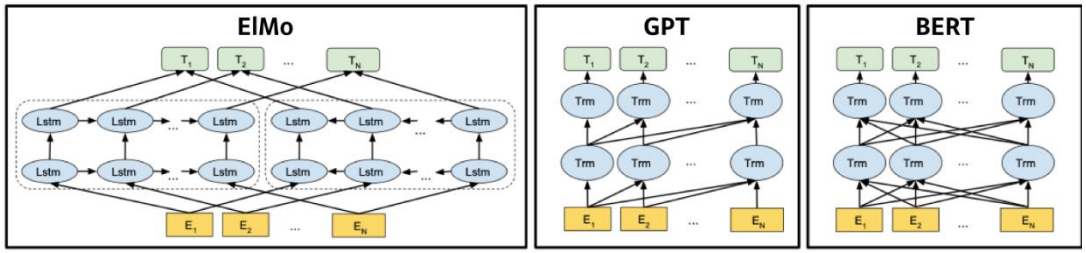
알고리즘 8-1 순환 신경망으로 문장 생성

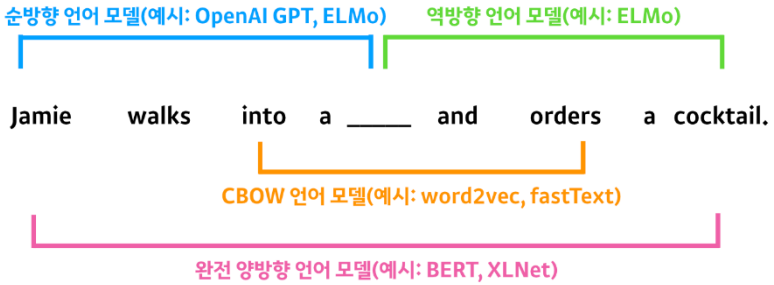
입력: 학습된 RNN 언어 모델
출력: 문장 $\mathbf{s}$

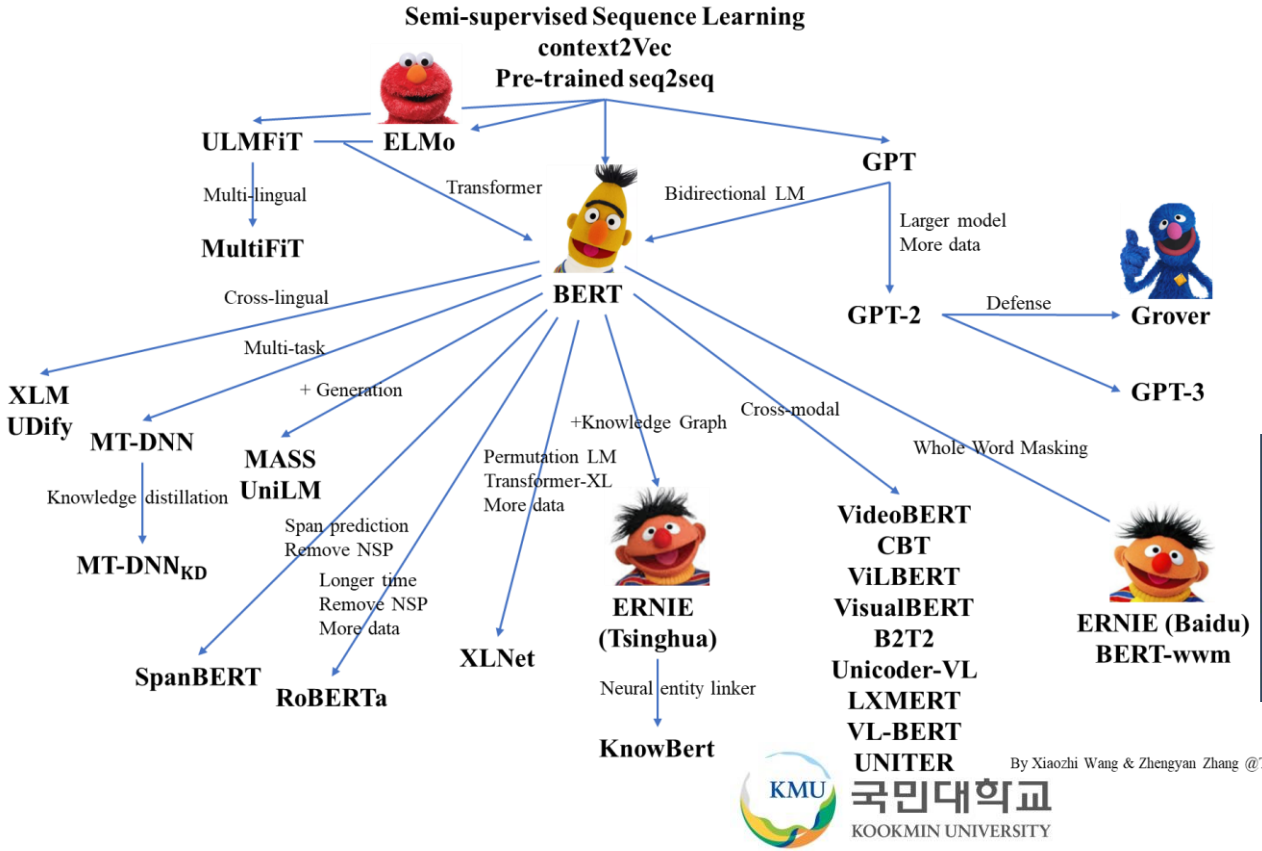
1 $\mathbf{s} = (< \text{시작} >)^{\mathrm{T}}$
2 while ($\mathbf{s}$의 마지막 요소 $\neq$ 〈끝〉)
3 $\mathbf{s}$를 RNN에 입력하여 출력 $\tilde{\mathbf{y}} = (\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_{|s|})^{\mathrm{T}}$를 구한다. // 다음 단어 예측
4 $\mathbf{w}_{|s|}$의 확률에 따라 m개 단어 중 하나를 샘플링한다.
5 $\mathbf{s}$의 끝에 라인 4에서 샘플링한 단어를 추가한다.

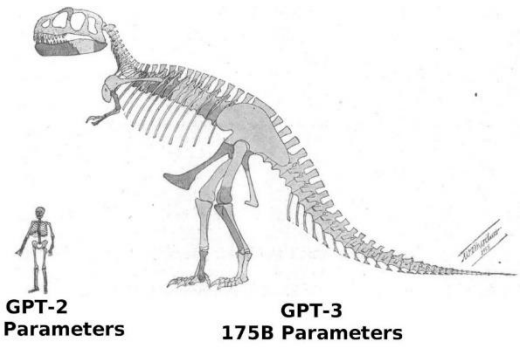
8.5.1 언어 모델

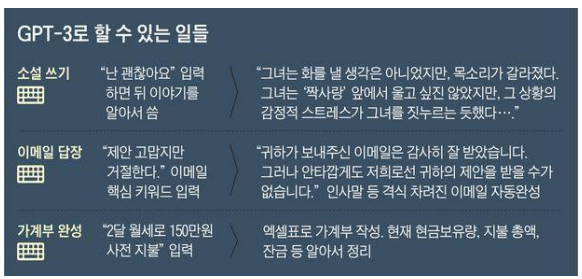
주요 언어 모델











KMU 국민대학교 KOOKMIN UNIVERSITY

8.5.2 기계 번역

기계 번역

- 훈련 샘플 예

$$\mathbf{x} = (< \text{시작} >, \text{자세히}, \text{보아야}, \text{예쁘다})^{\mathrm{T}}, \quad \mathbf{y} = (\mathrm{It}, \mathrm{is}, \mathrm{beautiful}, \mathrm{to}, \mathrm{see}, \mathrm{more}, \mathrm{closely}, < \text{끝} >)^{\mathrm{T}}$$

- 언어 모델보다 어려움
  - 언어 모델은 입력 문장과 출력 문장의 길이가 같은데,
    기계 번역은 길이가 서로 다른 열 대 열sequence to sequence 문제
  - 어순이 다른 문제

- 고전적인 통계적 기계 번역 방법의 한계 → 현재 심층학습 기반 기계 번역 방법이 주류

8.5.2 기계 번역

- LSTM을 사용하여 번역 과정 전체를 통째로 학습
  - LSTM 2개를 사용 (앞쪽은 부호기encoder, 뒤쪽은 복호기decoder)
  - 부호기는 원시 언어 문장 $\mathbf{x}$를 $\mathbf{h}_{Ts}$라는 특징 벡터로 변환
  - 복호기는 $\mathbf{h}_{Ts}$를 가지고 목적 언어 문장 $\mathbf{y}$를 생성함

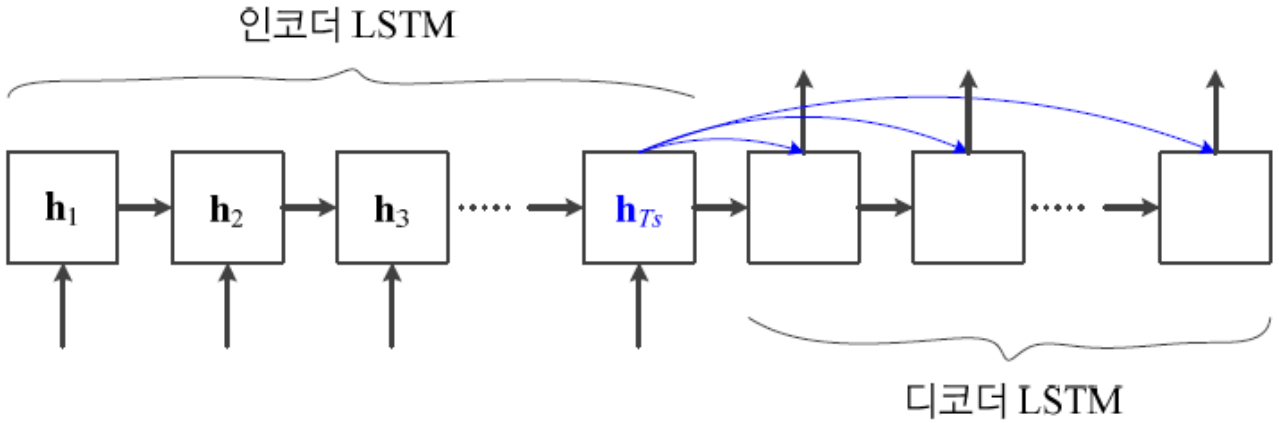


그림 8-21 인코더와 디코더가 특징 벡터를 하나만 사용하는 방식

  - 가변 길이의 문장을 고정 길이의 특징 벡터로 변환한 후, 고정 길이에서 가변 길이 문장을 생성
    → 문장이 길이가 크게 다를 때는 성능 저하

8.5.2 기계 번역

- 모든 순간의 상태 변수를 사용하는 방식
  - 부호기의 계산 결과인 $\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_{Ts}$를 모두 복호기에 넘겨 줌
  - 양방향 구조를 채택하여 어순이 다른 문제를 해결

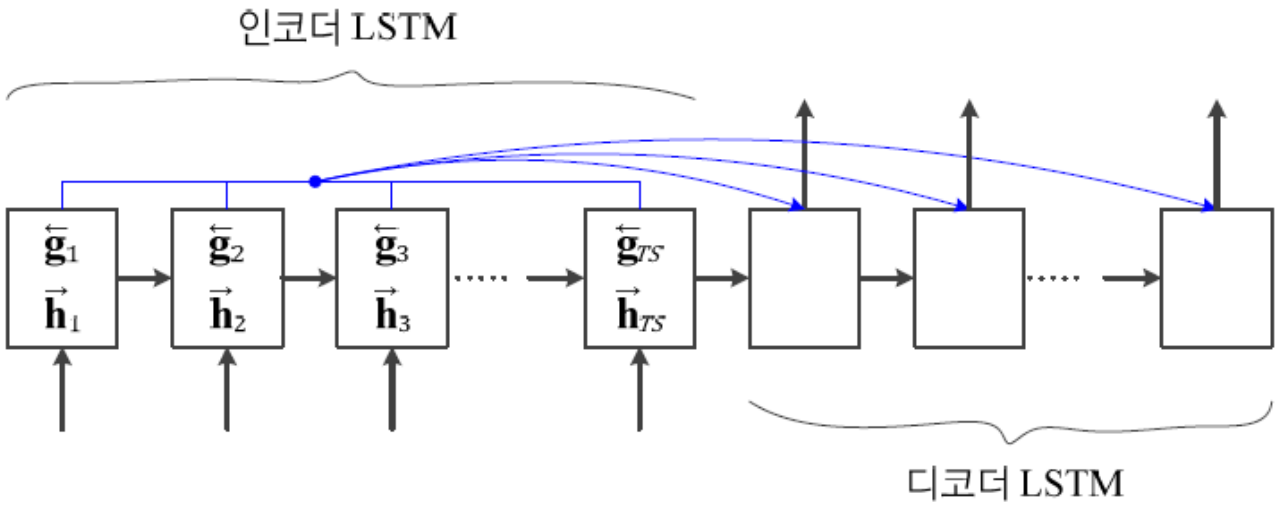


그림 8-22 인코더와 디코더가 여러 특징 벡터를 사용하는 방식

8.5.3 영상 주석 생성

- 영상 주석image caption 생성 데모 사이트

a man wearing a blue shirt with his arms on the grass.
a man holding a frisbee bat in front of a green field.
a man throwing a frisbee in a green field.
a boy playing ball with a disc in a field.
a young man playing in the grass with a green ball.

a red car on the side of the road in the small race.
a truck driving uphill on the side of the road.
a person driving a truck on the road.
a small car driving down a dirt and water.
a truck in a field of car is pulled up to the back.

a group of birds standing next to each other.
a group of ducks that are standing in a row.
a group of ducks that are standing on each other.
a group of sheep next to each other on sand.
a group of small birds is standing in the grass.

a kite flying over the ocean on a sunny day.
a person flying over the ocean on a sunny day.
a person flying over the ocean on a cloudy day.
a kite on the beach on the water in the sky.
a large flying over the water and rocks.

그림 8-24 영상 주석 자동 생성 데모 사이트[5]

8.5.3 영상 주석 생성

영상 주석 생성 응용

- 영상 속 물체를 검출하고 인식, 물체의 속성과 행위, 물체 간의 상호 작용을 알아내는 일 + 의미를 요약하는 문장 생성하는 일

  ← 매우 도전적인 문제

- 예전에는 물체 분할, 인식, 단어 생성과 조립 단계를 따로 구현한 후 연결하는 접근방법
- 현재는 딥러닝 기술을 사용하여 통째 학습

심층학습 접근방법

- CNN은 영상을 분석하고 인식 + LSTM은 문장을 생성

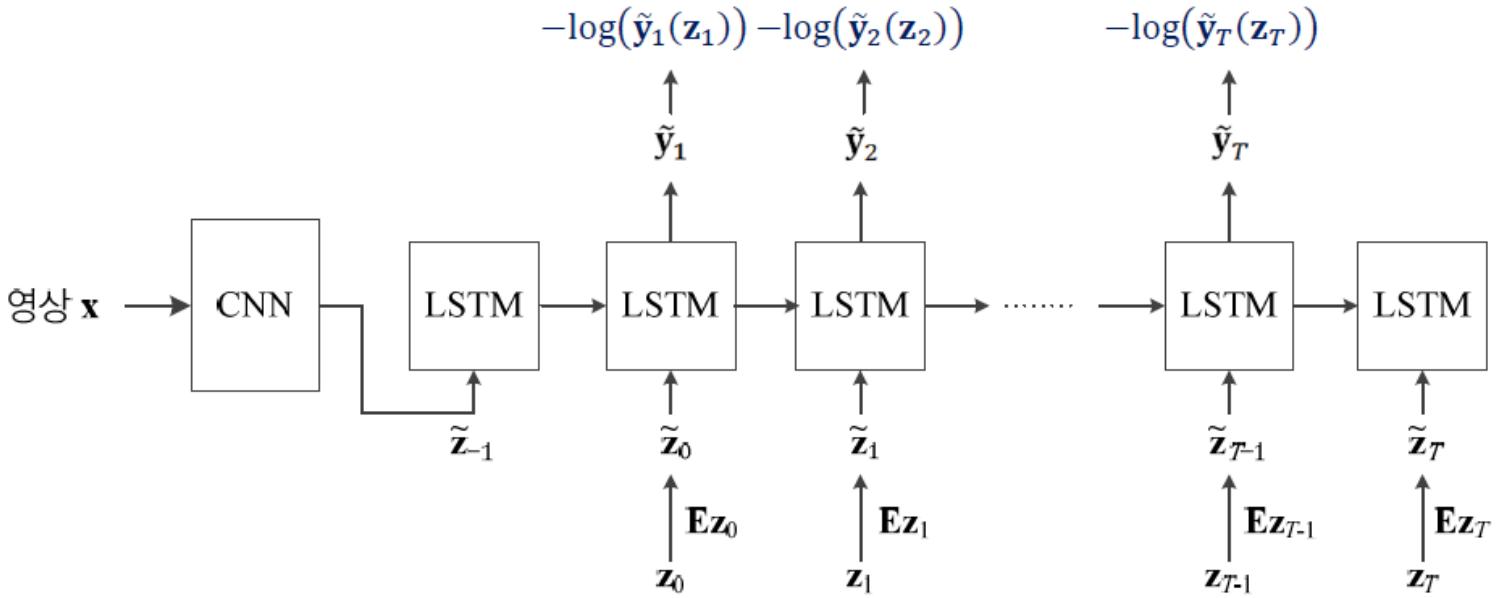


그림 8-23 자연 영상에서 자연어 문장을 생성하는 시스템 구조

8.5.3 영상 주석 생성

훈련집합

- $\mathbf{x}$는 영상, $\mathbf{y}$는 영상을 기술하는 문장 ($\mathbf{y} = (< \text{시작} >, \mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_T, < \text{끝} >)^{\mathrm{T}}$로 표현됨)

CNN

- 입력 영상 $\mathbf{x}$를 단어 임베딩 공간의 특징 벡터 $\tilde{\mathbf{z}}_{-1}$로 변환 (식 (8.50)의 첫 번째 줄)

$$\left.\begin{aligned} \tilde{\mathbf{z}}_{-1} &= cnn(\mathbf{x}) \\ \tilde{\mathbf{z}}_t &= \mathbf{E}\mathbf{z}_t, \quad t = 0,1,\cdots,T \end{aligned}\right\} \begin{aligned} &\text{: spatial feature learning} \\ &\text{: temporal feature learning} \end{aligned} \quad (8.50)$$

훈련 샘플 $\mathbf{y}$의 단어 $\mathbf{z}_t$는 단어 임베딩 공간의 특징 벡터 $\tilde{\mathbf{z}}_t$로 변환됨

- 식 (8.50)의 두 번째 줄에서 행렬 $\mathbf{E}$를 이용하여 변환
- $\mathbf{E}$는 통째 학습 과정에서 CNN, LSTM과 동시에 최적화됨

8.5.3 영상 주석 생성

- 학습 과정의 입력
  - 영상 $\mathbf{x}$를 CNN에 입력함
  - 문장 $\mathbf{z}_0, \mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_T$를 임베딩 공간의 점 $\tilde{\mathbf{z}}_0, \tilde{\mathbf{z}}_1, \tilde{\mathbf{z}}_2, \cdots, \tilde{\mathbf{z}}_T$로 변환하여 LSTM에 입력함

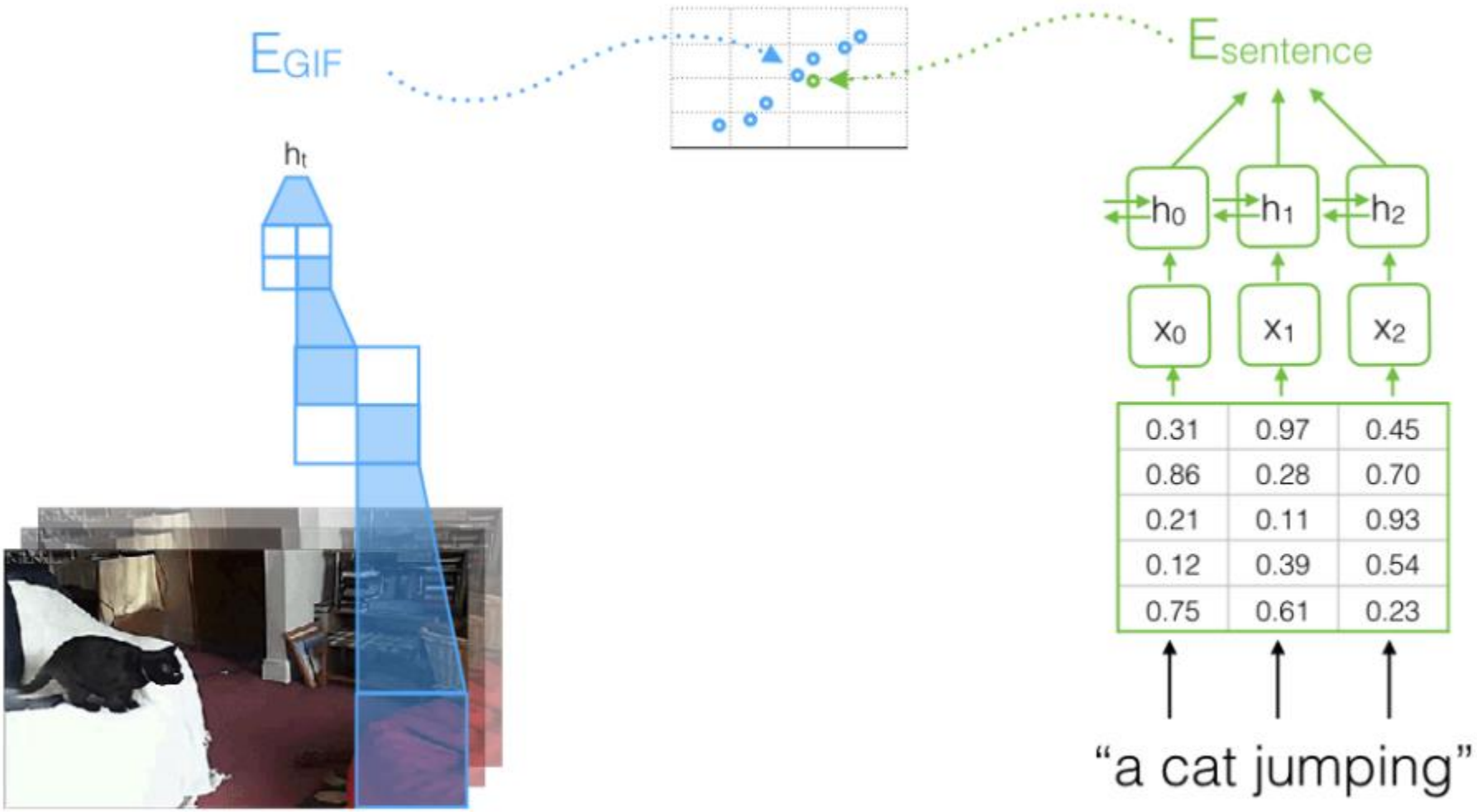
- 목적함수
  - LSTM의 출력 $\tilde{\mathbf{y}}_1, \tilde{\mathbf{y}}_2, \cdots, \tilde{\mathbf{y}}_T$와 $\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_T$가 일치할수록 예측을 잘한다고 평가
  - 식 (8.51)의 로그우도로 일치 정도를 평가

$$J(\Theta) = -\sum_{t=1}^{T} \log\left(\tilde{\mathbf{y}}_t(\mathbf{z}_t)\right) \qquad (8.51)$$

- 학습이 최적화해야 할 매개변수 집합
  - $\Theta = \{\text{CNN 매개변수},\ \text{LSTM 매개변수},\ \text{단어 임베딩 매개변수}\}$
    - 전이 학습을 사용하므로 CNN 매개변수는 완전연결층의 가중치
    - LSTM 매개변수는 식 (8.40)~(8.46)에 있는 $\mathbf{U}^g, \mathbf{W}^g, \mathbf{U}^i, \mathbf{W}^i, \mathbf{U}^o, \mathbf{W}^o, \mathbf{U}^f, \mathbf{W}^f$
    - 단어 임베딩 매개변수는 식 (8.50)의 $\mathbf{E}$
- Θ는 통째 학습으로 한꺼번에 최적화됨

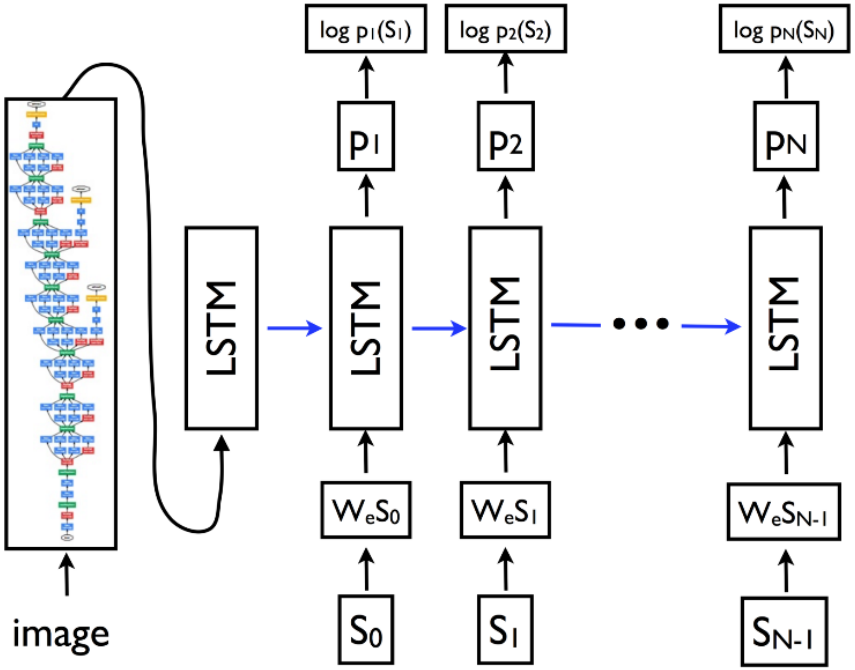
8.5.3 영상 주석 생성

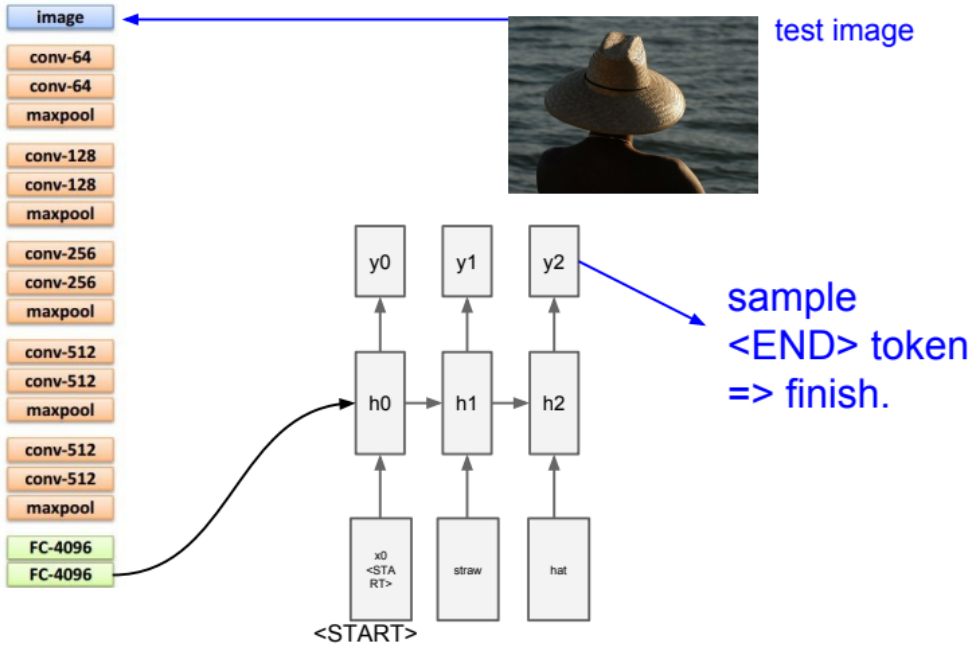
- 영상 주석 적용 사례



8.5.3 영상 주석 생성

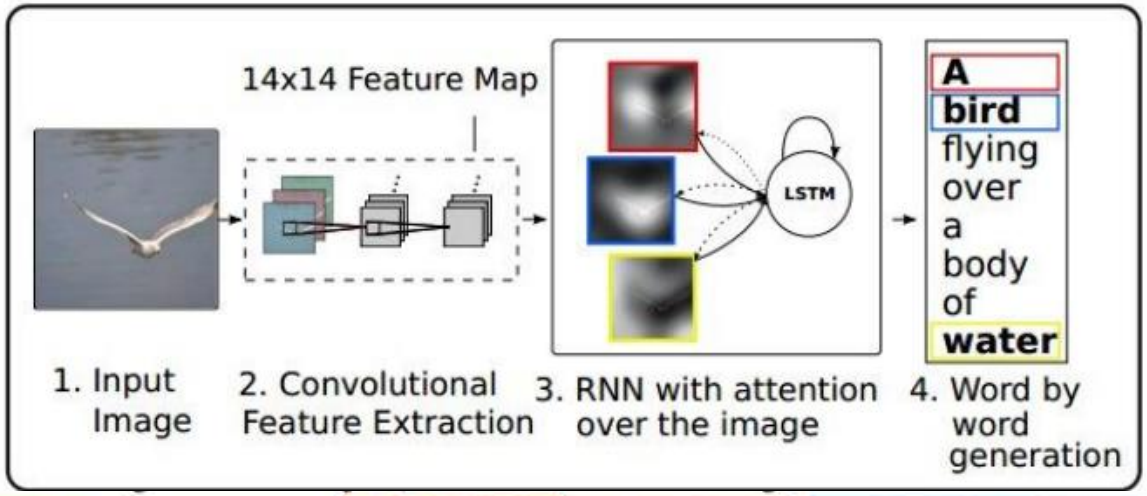
- 영상 주석 적용 사례





8.5.3 영상 주석 생성

- 영상 주석에서의 집중attention 적용 사례



A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

8.5.3 영상 주석 생성

- 영상 질의 응답 적용 사례

Q: What endangered animal is featured on the truck?

A: A bald eagle.
A: A sparrow.
A: A humming bird.
A: A raven.

Q: Where will the driver go if turning right?

A: Onto 24 ¾ Rd.
A: Onto 25 ¾ Rd.
A: Onto 23 ¾ Rd.
A: Onto Main Street.

Q: When was the picture taken?

A: During a wedding.
A: During a bar mitzvah.
A: During a funeral.
A: During a Sunday church service

Q: Who is under the umbrella?

A: Two women.
A: A child.
A: An old man.
A: A husband and a wife.