시퀀스 모델링

7장

contents

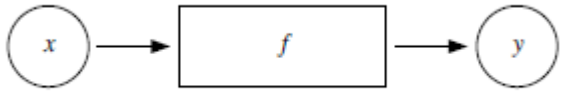
- 순환 신경망(RNN)
  - 피드포워드
  - BPTT
  - 기울기 소실
  - 여러 계층을 갖는 RNN
  - 양방향 RNN
- LSTM
- GRU
- 그래디언트 클리핑

7.1 들어가며

- 문제 해결에 시간(순서 정보)의 개념을 적용
  - 주식시장의 주가예측
  - 일기예보
  - 음성인식
  - 번역
- 자연어처리에 적용
  - 문장 내 단어들은 앞뒤 위치에 따라 서로 영향을 미침
  - 문서 내 문장들도 마찬가지
  - 순차적으로 입력을 넣음
  - 입력에 따라 모델의 은닉 상태가 순차적으로 변화
  - 상태에 따라 출력 결과가 순차적으로 반환
- 시퀀셜 모델링 (sequential modeling)
  - 은닉 마르코프 모델, 조건부 랜덤 필드
  - 신경망에서는 순환 신경망 (RNN) 사용

7.2 순환 신경망

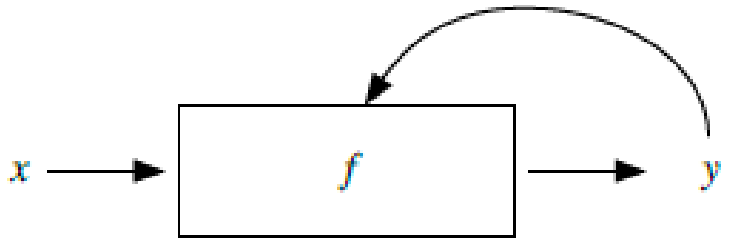
- 기존 : $y = f(x)$



▶ 기존 신경망 구조

- 순환 신경망(RNN)은 입력 x_t와, 직전의 은닉 상태hidden state인 h_{t-1}를 참조하여 현재의 상태인 h_t를 결정하는 작업을 여러 time-step에 걸쳐 수행

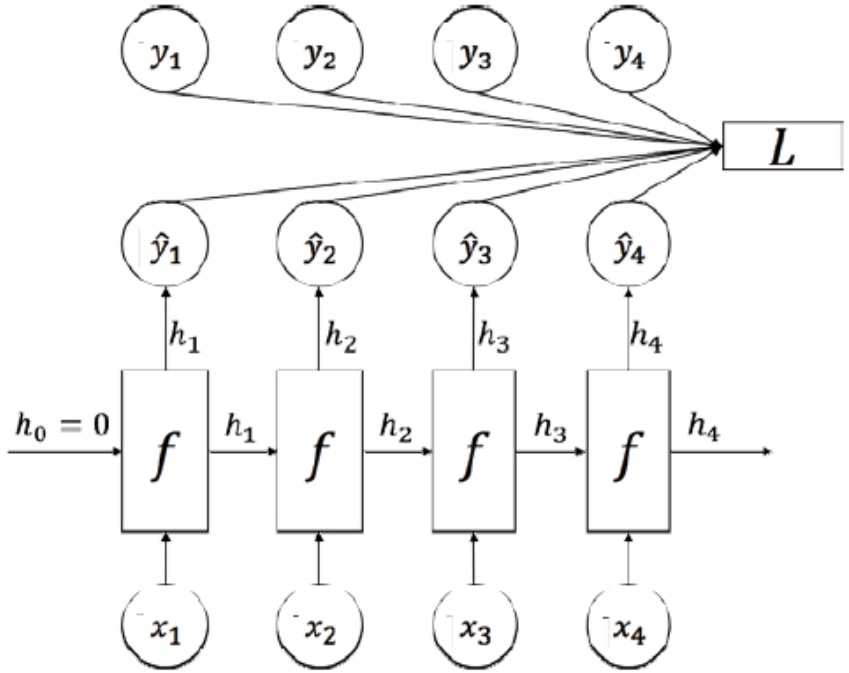
$$h_t = f(x_t, h_{t-1}; \theta)$$



▶ 재귀적 속성이 반영된 뉴럴넷 구조

7.2.1 값이 앞으로 전달되는 과정 : 피드포워드

- 피드 포워드
  - 매 time-step마다 은닉 상태를 활용해 손실값을 계산



$$\hat{y}_t = h_t = f(x_t, h_{t-1}; \theta)$$
$$= \tanh(W_{ih}x_t + b_{ih} + W_{hh}h_{t-1} + b_{hh})$$
$$\text{where } \theta = \{W_{ih}, b_{ih}, W_{hh}, b_{hh}\}$$

$$x_t \in \mathbb{R}^w,\ h_t \in \mathbb{R}^d,\ W_{ih} \in \mathbb{R}^{d \times w},\ b \in \mathbb{R}^d,\ W_{hh} \in \mathbb{R}^{d \times d},\ b_{hh} \in \mathbb{R}^d$$

$$\mathcal{L} = \frac{1}{n}\sum_{t=1}^{n}\mathcal{L}(\hat{y}_t, y_t)$$

RNN의 입력 텐서와 은닉 상태 텐서의 크기

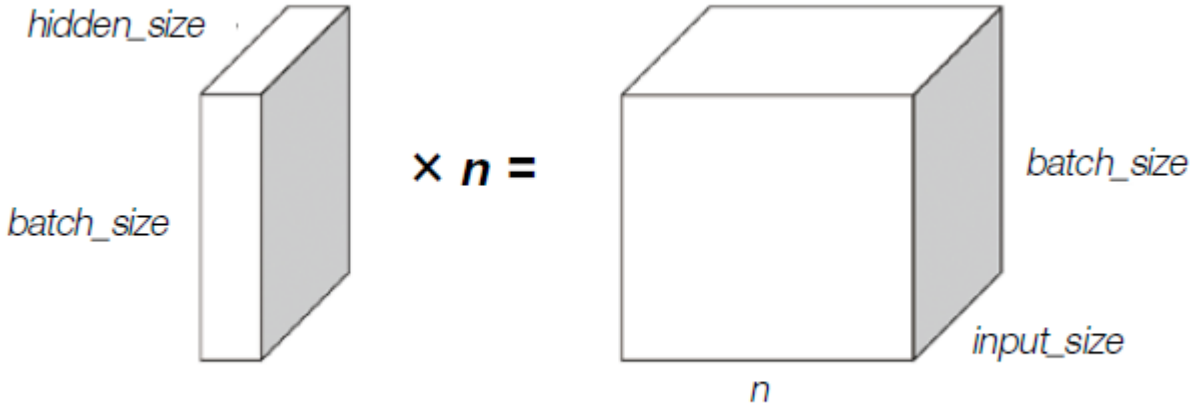
- 피드 포워드
  - RNN의 입력 텐서와 은닉 상태 텐서의 크기
    - 입력으로 주어지는 x의 미니배치까지 감안한 크기

$$x_t \in \mathbb{R}^{\text{batch\_size} * 1 * \text{input\_size}}$$

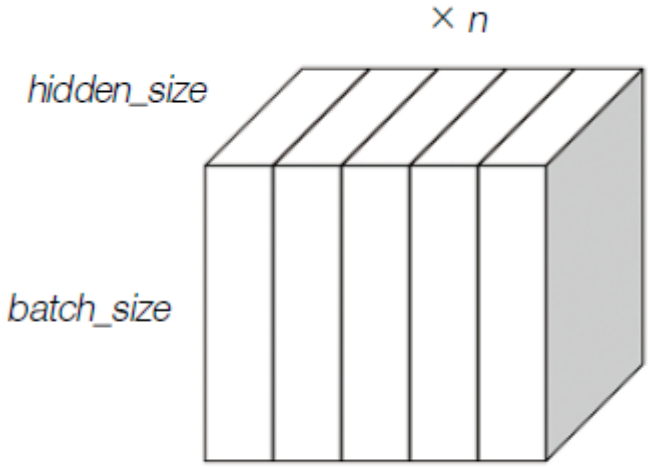
$$|h_{1:n}| = (\text{batch\_size}, n, \text{hidden\_size})$$

$$\text{where } h_{1:n} = [h_1; h_2; \cdots; h_n]$$

$$|h_t| = (\text{batch\_size}, \text{hidden\_size})$$



(batch_size, 1, input_size) × n = (batch_size, n, input_size)



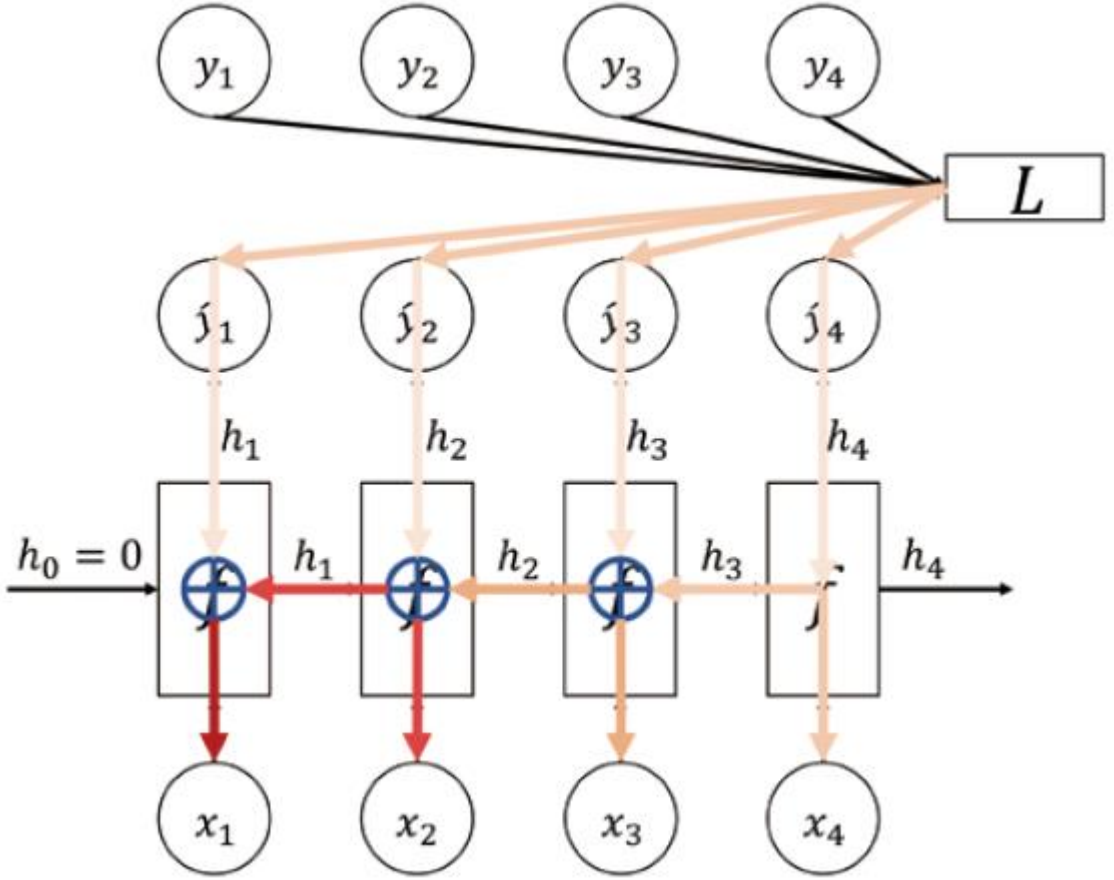
(batch_size, n, hidden_size)

7.2.2 BPTT

- BPTT (Back-propagation Through Time)
  - RNN에 사용된 파라미터 θ는 모든 시간에 공유되어 사용됨

$$\frac{\partial \mathcal{L}}{\partial \theta} = \sum_t \frac{\partial \mathcal{L}(y_t, \hat{y}_t)}{\partial \theta}$$

Time-step 이 길어짐에 따라
매우 깊은 신경망과 유사하게 동작



7.2.3 기울기 소실

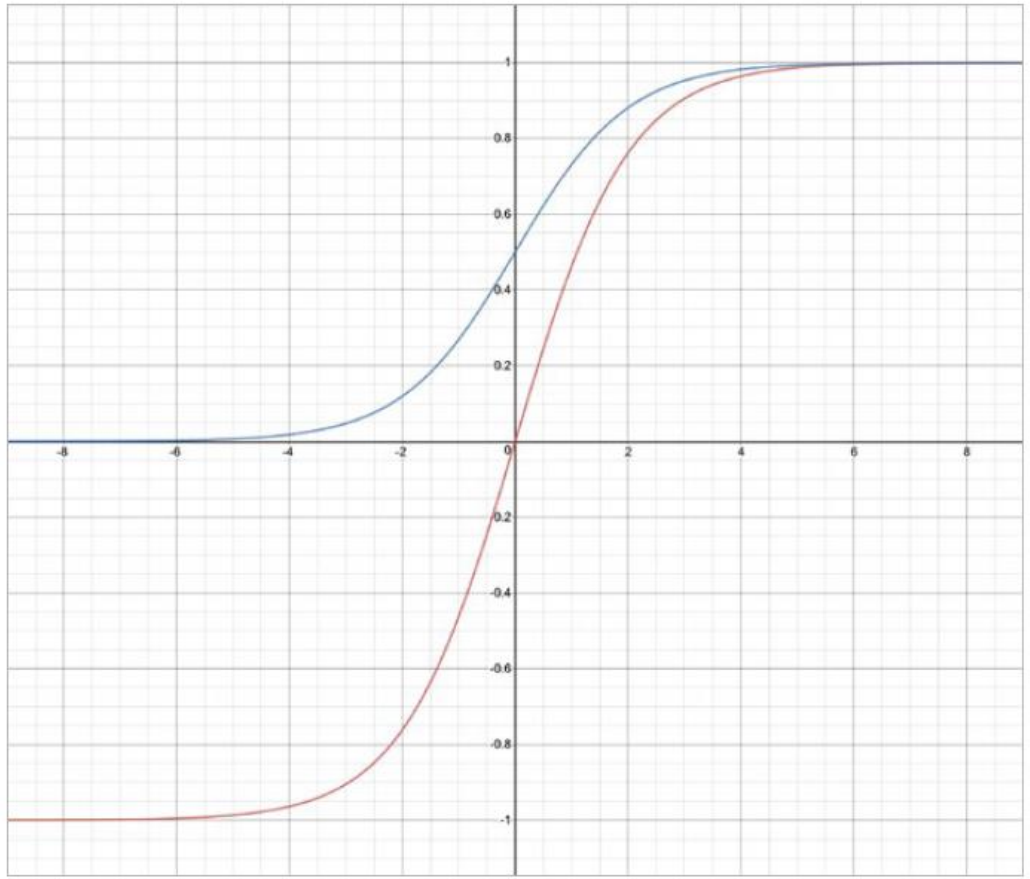
- Time-step 만큼의 계층이 있는 것과 비슷
- 활성화 함수로 tanh 함수 사용

$$tanh(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$$

$$sigmoid(x) = \frac{1}{1 + e^{-x}}$$

$$= 2 \times tanh(2x) - 1$$

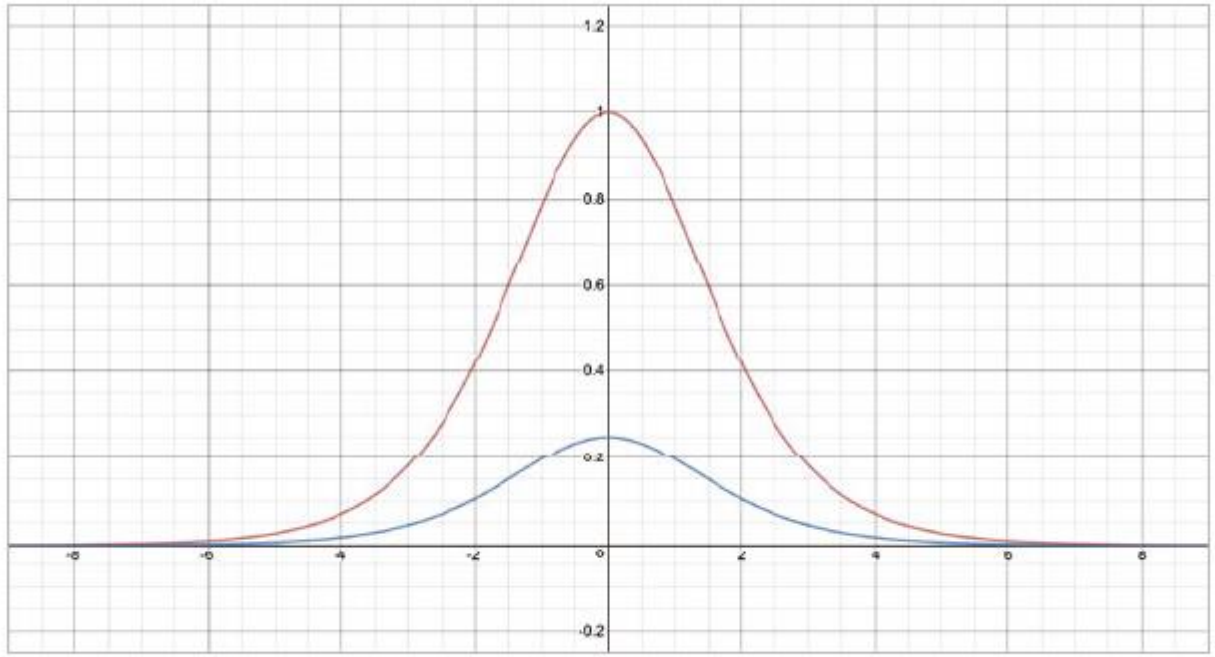
  - 양 끝의 tanh(x) 값이 -1 또는 1에 근접
  - 양 끝의 기울기가 0에 가까와짐



▶ 빨간색: tanh 함수, 파란색: sigmoid 함수

7.2.3 기울기 소실

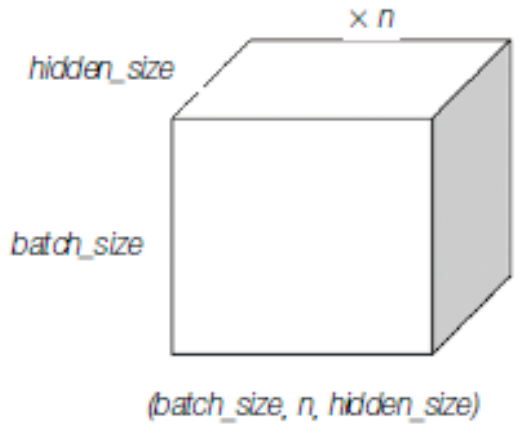
- 기울기 소실 (vanishing gradient)
  - 층을 거칠수록 기울기가 작아짐
  - RNN 또는 DNN 에서 쉽게 발생
  - DNN에서는 ReLu 와 레지듀얼 커넥션으로 해결



▶ 빨간색: tanh의 도함수, 파란색: sigmoid의 도함수

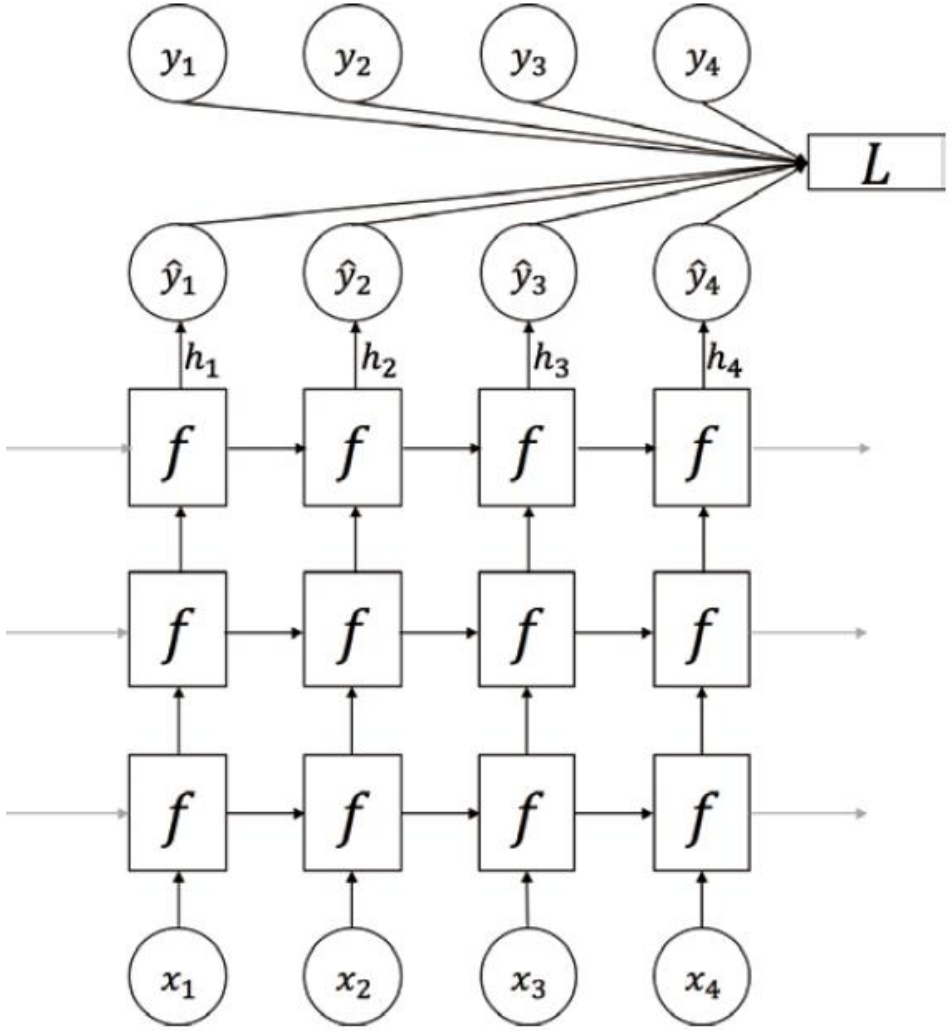
7.2.4 여러 계층을 갖는 RNN

- 하나의 time-step 내에서 여러 층의 RNN을 쌓을 수 있음
- 각 time-step의 RNN 전체 출력값은 맨 위층의 은닉 상태
- 출력 텐서의 크기



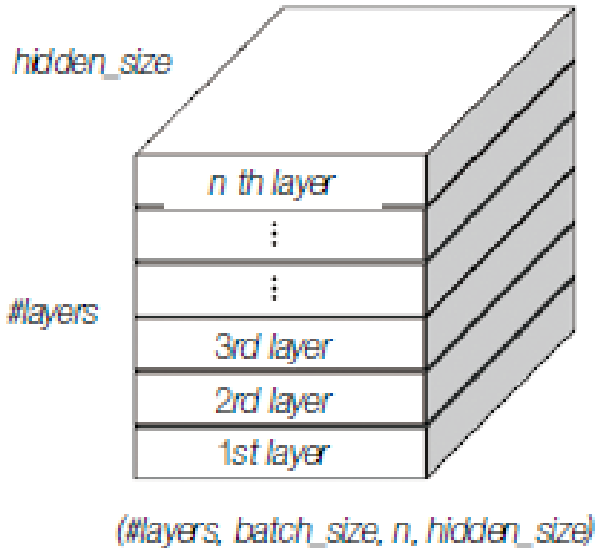
▶ RNN의 출력 텐서(n time-step)

$$|h_{1n}| = (\text{batch\_size}, n, \text{hidden\_size})$$



7.2.4 여러 계층을 갖는 RNN

- 여러 계층을 갖는 RNN의 은닉 상태의 크기

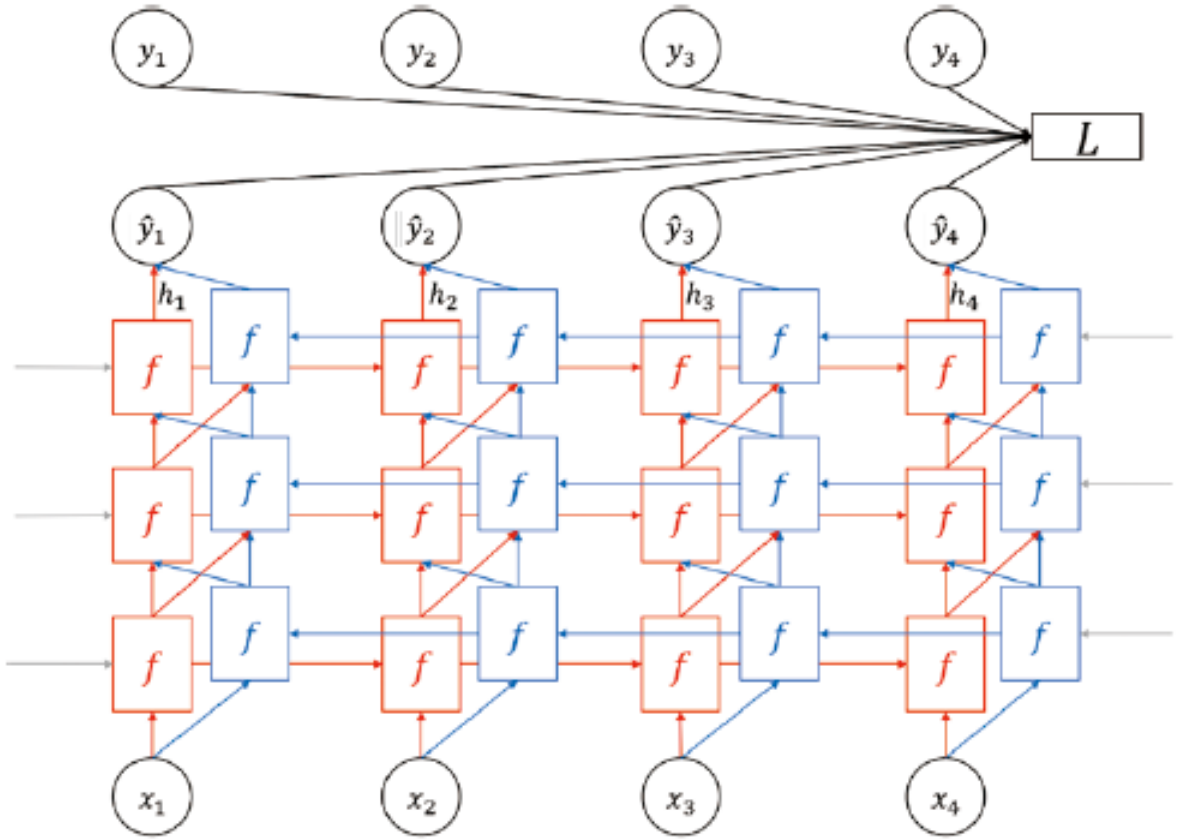


▶ 여러 층을 가진 RNN의 은닉 상태(1 time-step)

$$|h_t| = (\#\text{layers},\ \text{batch\_size},\ \text{hidden\_size})$$

7.2.5 양방향 RNN

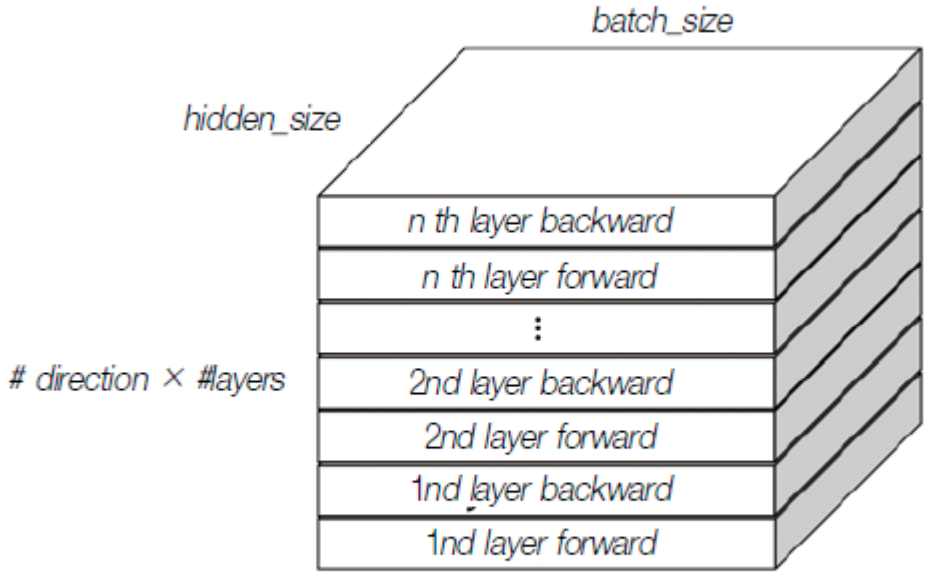
- 양방향 RNN
  - 기존의 정방향에 역방향이 추가
  - 마지막 time-step에서부터 거꾸로 입력
  - 정방향과 역방향의 파라미터는 공유되지 않음



▶ 두 방향으로 은닉 상태를 전달 및 계산하는 RNN의 형태

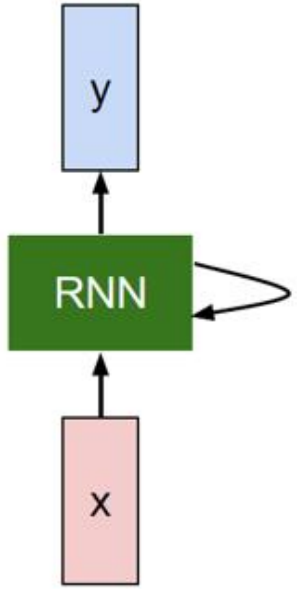
7.2.5 양방향 RNN

- 여러 층의 양방향 RNN
  - 각 층마다 두 방향의 time-step별 은닉 상태 값을 이어붙여서 다음 층의 방향별 입력
  - 가운데 일부 층만 양방향 RNN 층을 사용
  - 양방향 RNN의 은닉 상태 텐서의 크기

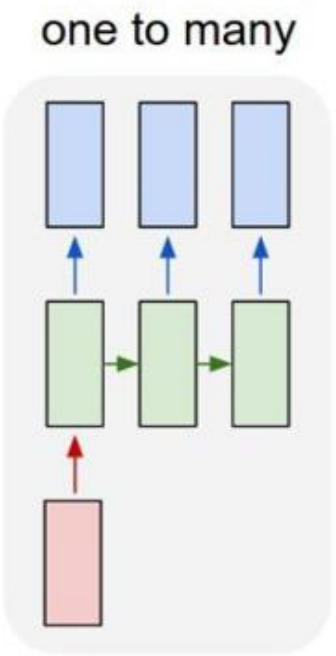


$$|h_t| = (\#\,\text{direction} \times \#\,\text{layers}, \text{batch\_size}, \text{hidden\_size})$$

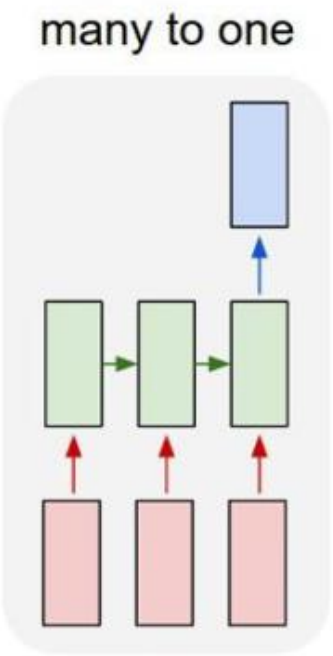
7.2.6 자연어 처리에 RNN을 적용하는 사례



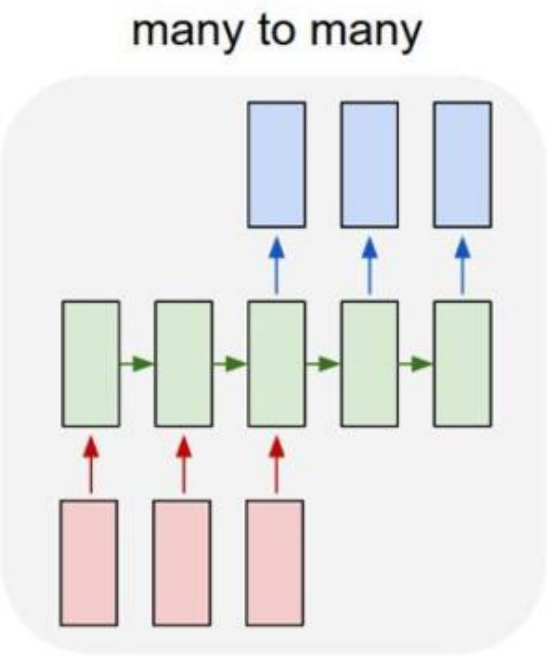
<기본구조>



<사진설명 붙이기>
사진 → 단어들



<감성분석>
단어들 → 감성점수



<번역>
단어들 → 단어들

7.2.6 자연어 처리에 RNN을 적용하는 사례

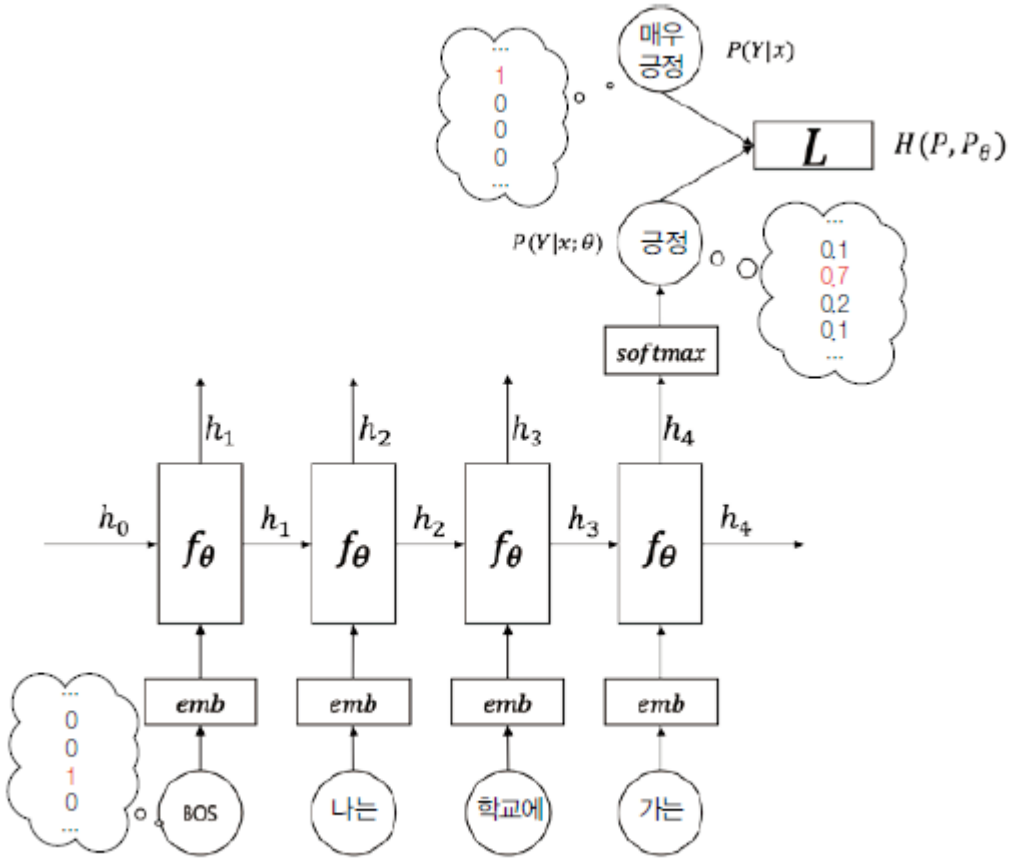
- 하나의 출력
  - softmax 함수를 통해 해당 입력 텍스트의 확률 분포를 근사하도록 동작합니다.

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{|C|} e^{x_j}}$$

$$\text{CrossEntropy}(y, \hat{y}) = -\sum_{i=1}^{|C|} y_i \log \hat{y}_i$$

where y and $\hat{y}$ is probability distribution, such as $\hat{y} = P(\text{y} \mid x; \theta)$

thus, $y_i = P(\text{y} = i \mid x)$ and $\hat{y}_i = P(\text{y} = i \mid x; \theta)$

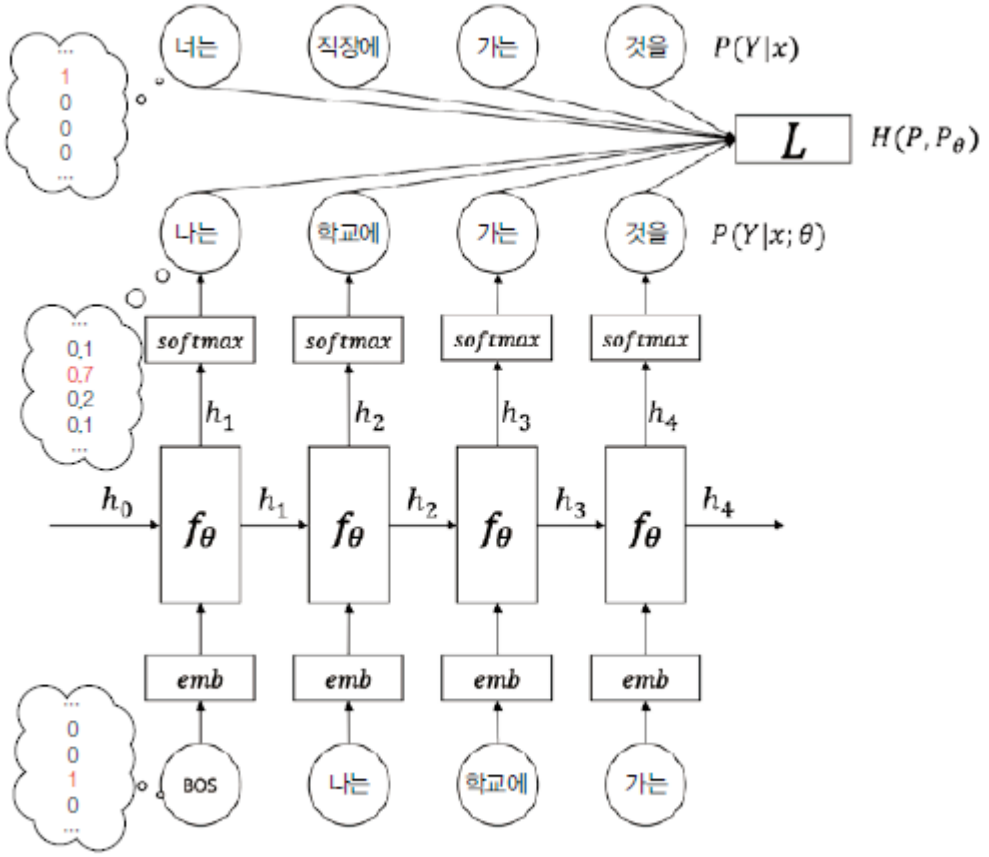


▶ RNN의 마지막 time-step의 출력을 사용할 경우

7.2.6 자연어 처리에 RNN을 적용하는 사례

- 모든 출력을 사용할 경우
  - 언어 모델, 번역, 형태소 분석 등

- 자기회귀 모델 (autoregressive model)
  - 이전 자신의 상태가 현재 자신의 상태를 결정
  - 양방향 RNN을 사용할 수 없음



▶ 모든 time-step의 출력을 사용할 경우

7.2.7 정리

- 자연어처리는 대부분 분류 문제
  - 입출력 형태가 모두 불연속적인 값
  - 교차 엔트로피 손실 함수를 사용하여 신경망을 학습
- RNN (recurrent neural network)
  - 가변 길이의 입력을 받아 가변길이의 출력을 내어줌
  - Time-step이 길어질수록 앞의 데이터를 기억하지 못함
  - LSTM, GRU 등으로 해결