

딥러닝 입문 (RNN)

목 차

01 RNN 기본 이해

02 RNN의 활용

03 RNN 기본 구조의 이해

04 RNN 기본 구조의 이해(2)

05 [실습] 숫자 데이터 예측

06 [실습] 단어 자동 완성 RNN 모델

07 구글 번역에 사용된 모델

08 LSTM 모델

09 Transformer 모델

01 RNN 기본 이해

Recurrent Neural Networks(RNN) : 순환 신경망이라 말한다.

인공 신경망의 하나의 종류로서, 유닛 간의 연결이 순환적 구조를 갖는다.

02 RNN은 어디에 쓰일까?

(1) 자연어 처리 - 기계 번역, 감성 분석, 텍스트 생성, 챗봇

A. 하나의 예로 구글 뉴럴 기계 번역기(2016년도)

- 기존의 성능을 크게 뛰어 넘음
- 몇몇 언어에서는 인간에 가까운 수준까지 도달
- 전체 구성은 Encoder-Decoder로 이루어짐.
- Encoder와 Decoder는 각각 8층의 순환신경망으로 구성됨.
(순환 신경망의 기본 모듈로는 LSTM을 사용)

*** 자연어 처리 분야에서는 2024년 9월 현재 Transformer 모델이 거의 많이 사용되고 있음.**

참조 링크 : <https://ai.googleblog.com/2016/09/a-neural-network-for-machine.html>

02 RNN은 어디에 쓰일까?

(1) 텍스트 분류 - 감정 분석

A. 브랜드 매니지먼트 - 다양한 고객층의 브랜드에 대한 일반적 인식을 추적하기 위해 정서를 분석

B. 시장 조사 - 어떤 기술과 청중의 반응과 관련된 정보를 수집

C. 제품 분석 - 제품 또는 제품의 특정 측면에 대한 모든 종류의 고객 의견을 관리하고 분석하여 추가 개선 계획을 수립

* 자연어 처리 분야에서는 2024년 9월 현재 Transformer 모델이 점점 더 많이 사용되고 있음.

02 RNN은 어디에 쓰일까?

(2) 필기체 인식

- A. 필기체를 입력하면 정확한 글자로 출력이 되어 나타난다.
- B. 정자를 넣으면 필기체로 출력이 되어 나타난다.
- C. 실제 아무것도 넣지 않아도 필기체가 생성되기도 함.

(3) 음성 인식기

- A. 하나의 딥러닝 모델로 영어와 중국어를 모두 인식함.

02 RNN은 어디에 쓰일까?

(4) 영상 분석

- A. 동작 인식(Action Recognition) : 영상 내에서 사람이나 객체의 동작을 시간 순서대로 발생하는 데이터를 분석. (예) RNN은 다리의 움직임, 팔의 흔들림, 머리의 움직임 등을 시간 순서대로 분석하여 '걷기' 라는 동작을 추론
- B. 객체 추적(Object Tracking) : RNN은 영상 내에서 특정 객체를 추적하는 데 사용.
- C. 영상 생성(Video Summarization) : RNN은 긴 영상을 요약하는 데 사용.
- D. 영상 캡션 생성(Video Captioning) : RNN은 영상에 대한 텍스트 캡션을 생성하는 데 사용.

02 RNN은 어디에 쓰일까?

- 영상 캡션 생성

A. CNN과 RNN을 조합하여 이미지의 캡션을 자동으로 생성한다.



참조

<https://github.com/danieljl/keras-image-captioning>

02 RNN은 어디에 쓰일까?

(5) 시계열 예측

A. **주식 예측** : 주식 시장의 과거 데이터를 바탕으로 미래의 가격 변동 예측.

B. **날씨 예보** : 기상 데이터를 기반으로 특정 지역의 날씨 패턴을 예측하는데 활용.

C. **전력 수요 예측**

D. **센서 데이터 분석**

02 RNN은 어디에 쓰일까?

RNN은 시계열 데이터나 자연어 처리와 같이 시간의 흐름이나 문맥이 중요한 데이터를 처리하는 데 적합한 모델로, 이러한 분야에서 활발하게 활용되고 있습니다.

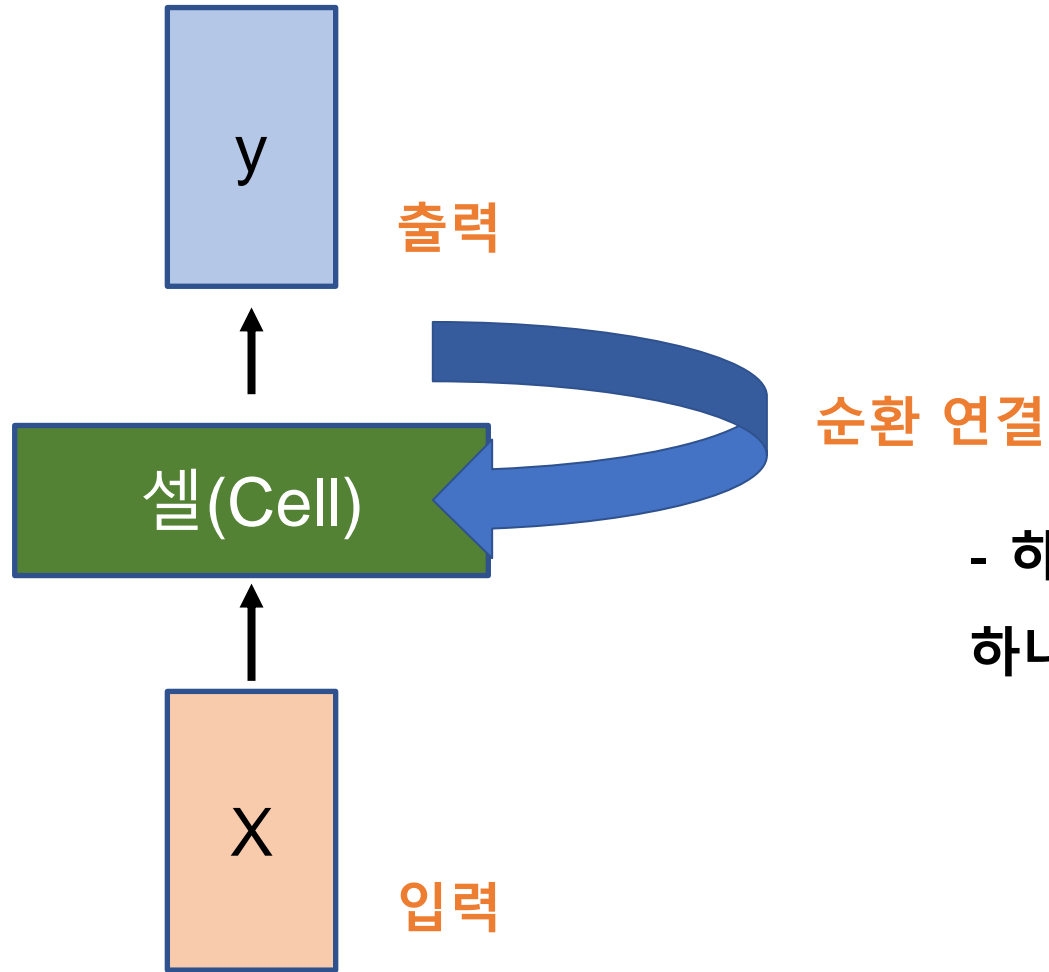
시계열 데이터나 자연어 처리에 다음과 같은 모델도 사용되고 있음.

LSTM, GRU, Transformer, CNN, TCN(시간 합성곱 네트워크), ARIMA, Prophet, XGBoost, LightGBM, Bi-RNN

02 RNN 모델과 Transformer 모델

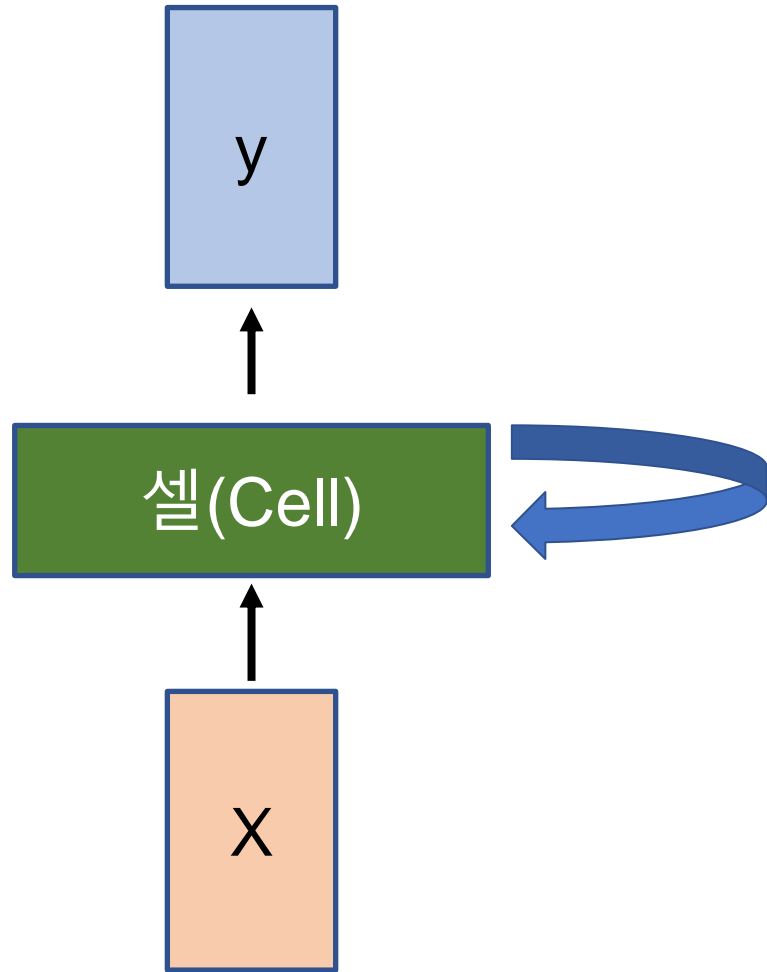
- 특정 도메인에서 강점을 보이고 있다. (시계열 예측 문제, 실시간 처리 시스템)
데이터가 순차적으로 들어오는 경우나, 실시간 처리가 필요한 경우는 여전히 강력함.
- 작은 규모의 데이터 셋이나 제한된 컴퓨팅 리소스 환경에서 RNN이 상대적으로 효율적이다. (적은 메모리 사용량으로 임베디드 시스템이나 모바일 기기에 유리)
- 하이브리드 모델이 등장(RNN과 Transformer의 장점 결합한 모델 등장)
- RNN의 전망은 축소되고 있지만, 소멸되지는 않을 것으로 예상.

03 RNN 기본 구조의 이해



- 하나의 셀의 결과가 다음 셀의 입력의 하나로 들어가 적용된다.

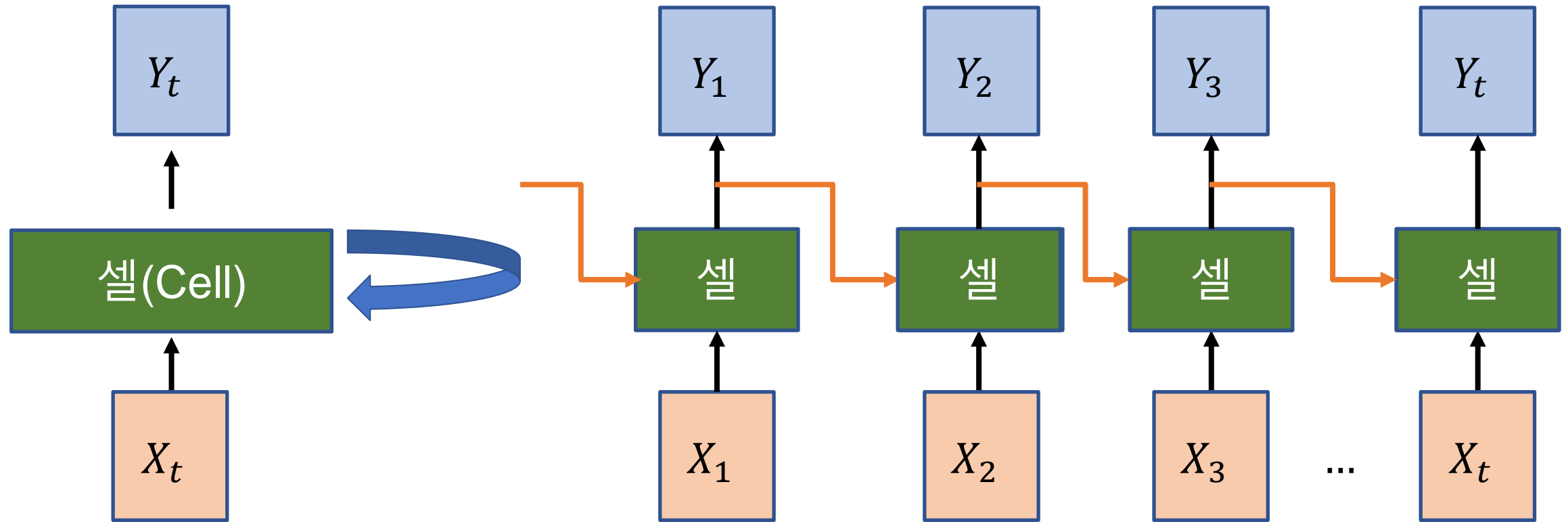
03 RNN 기본 구조의 이해



* Key point

- RNN은 셀을 여러 개 중첩하여 심층 신경망을 만든다.

03 RNN 기본 구조의 이해 - Many-to-Many



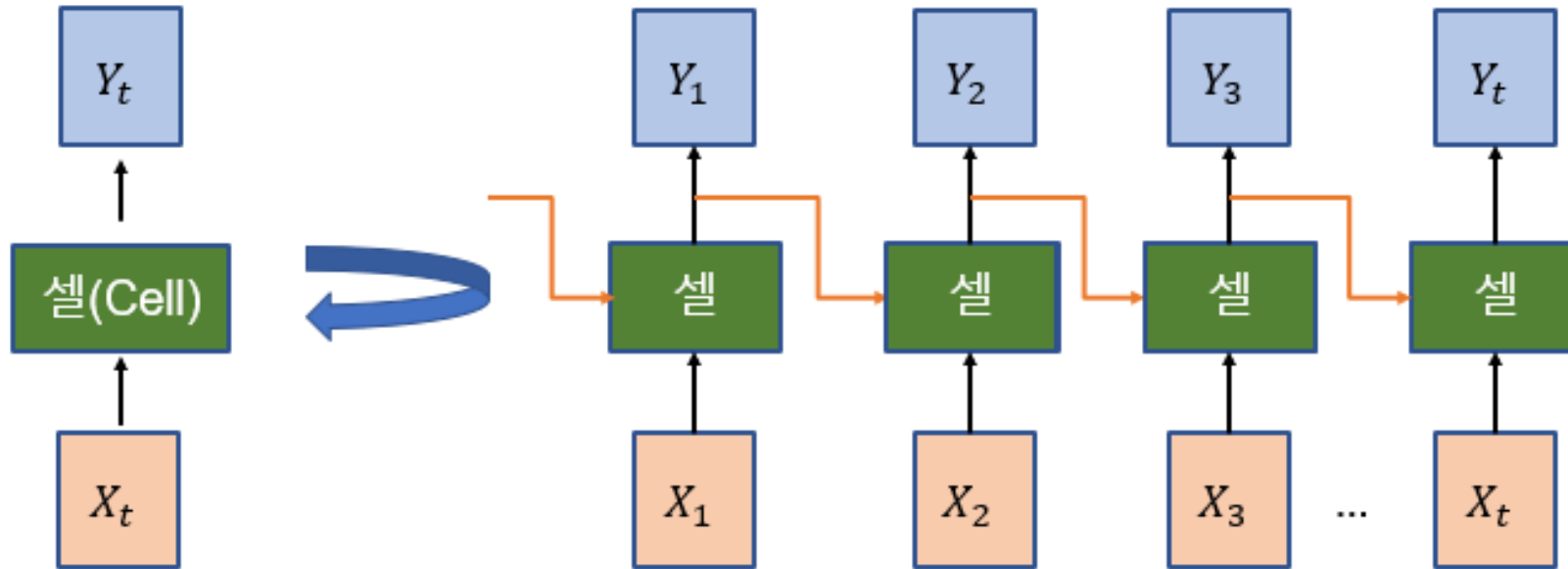
* Key point

- 앞단계에서 학습한 결과를 다음 단계의 학습에 이용.
- 학습 데이터를 단계별로 구분하여 입력을 하여야 함.

$(X_1, X_2, X_3, \dots, X_t)$

- (활용 예) 기계 번역: 입력 문자의 각 단어가 대응되는 다른 언어로 번역

03 RNN 기본 구조의 이해 - Many-to-Many

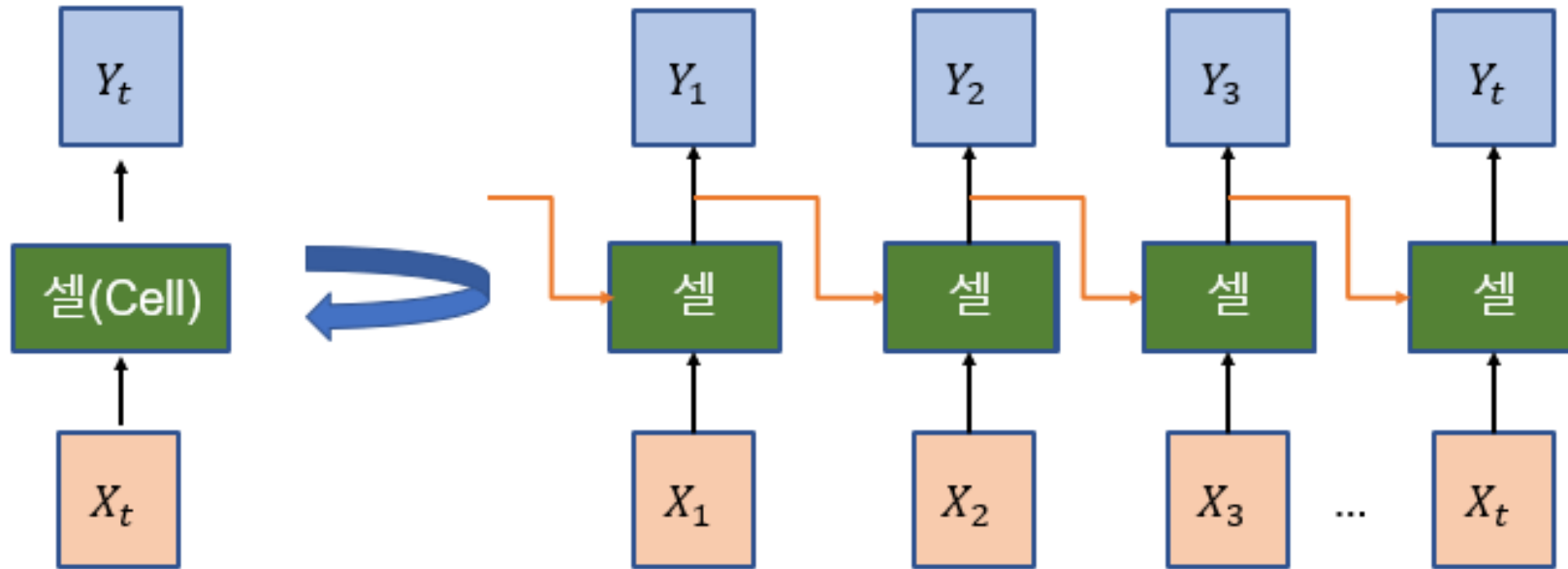


01 RNN 신경망은 순서가 있는 데이터(시퀀스 데이터)를 다룬다.

02 RNN 기본 신경망은 긴 단계의 경우,

맨 뒤에서 맨 앞의 정보를 잘 기억하지 못하는 특성이 있음.

03 RNN 기본 구조의 이해 – Many-to-Many



01 RNN 신경망은 순서가 있는 데이터를 다룬다.

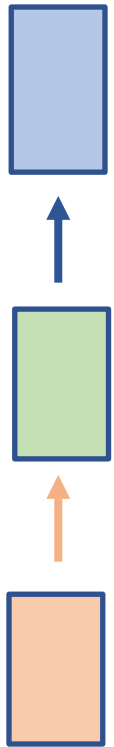
02 RNN 기본 신경망은 긴 단계의 경우,

맨 뒤에서 맨 앞의 정보를 잘 기억하지 못하는 특성이 있음.

제안 : LSTM – Long Short-Term Memory, GRU : Gated Recurrent Units

03 RNN 기본 구조의 이해 - 다양한 RNN 모델

one to one



RNN

one to many

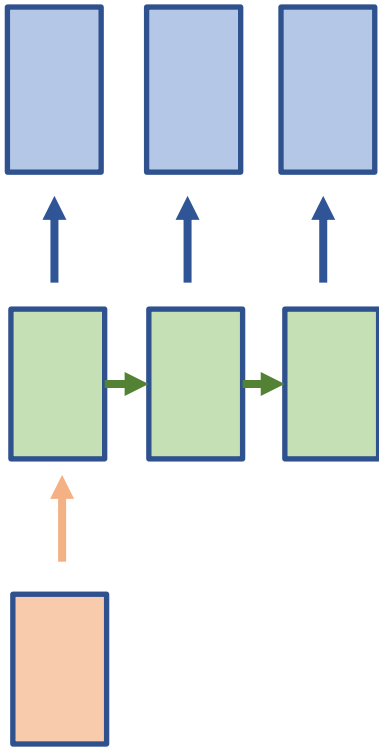
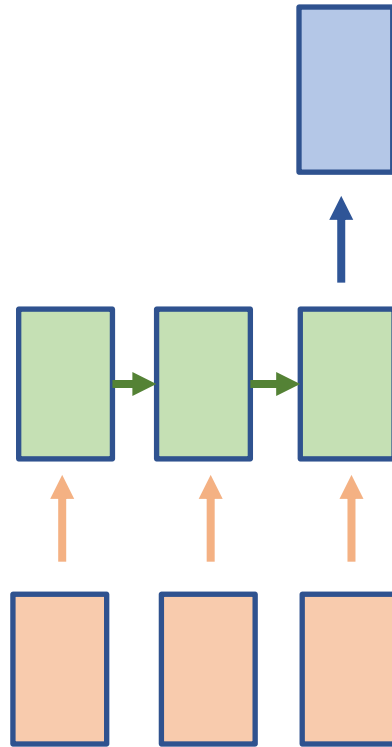


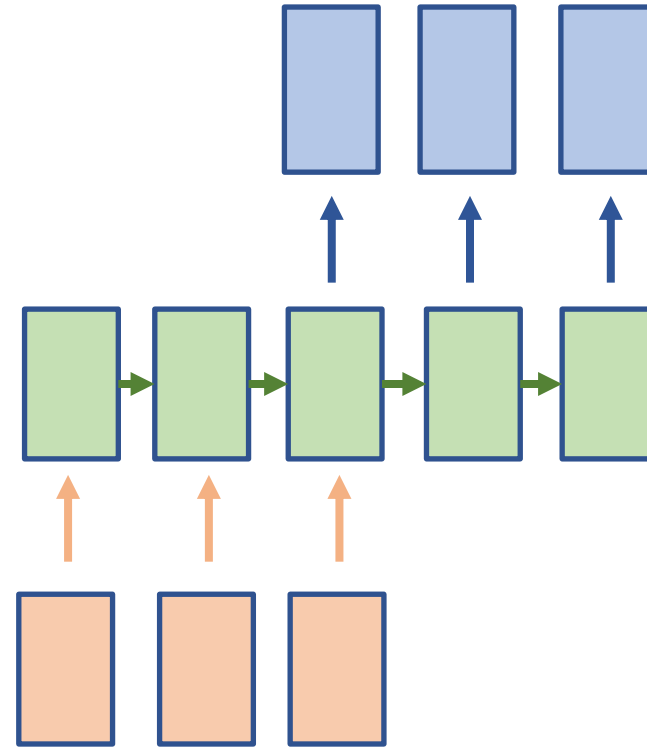
Image
Captioning

many to one



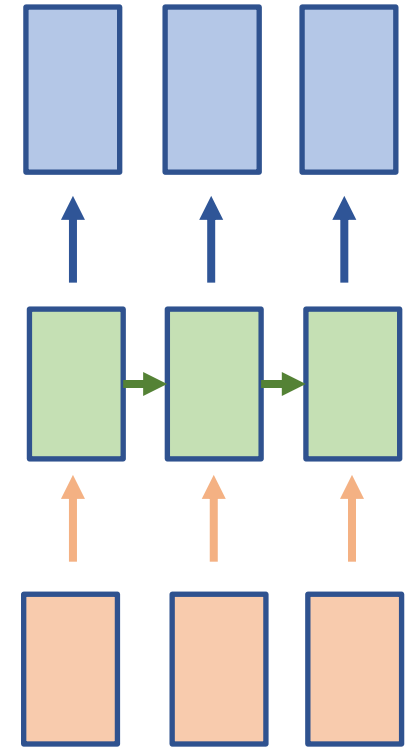
Sentiment
Classification

many to many



Machine Translation

many to many



Video
Classification

04 RNN 기본 구조의 이해(2) - 기본 용어(시퀀스)

사전에서는 이렇게 말한다.

- (1) (일련의)연속적인 사건들,
- (2) (사건 행동등의) 순서
- (3) (영화에서 연속성 있는 하나의 주제 정경으로 연결되는)장면

신경망에서는 이렇게 말한다.

Sequence of words (단어들의 시퀀스)

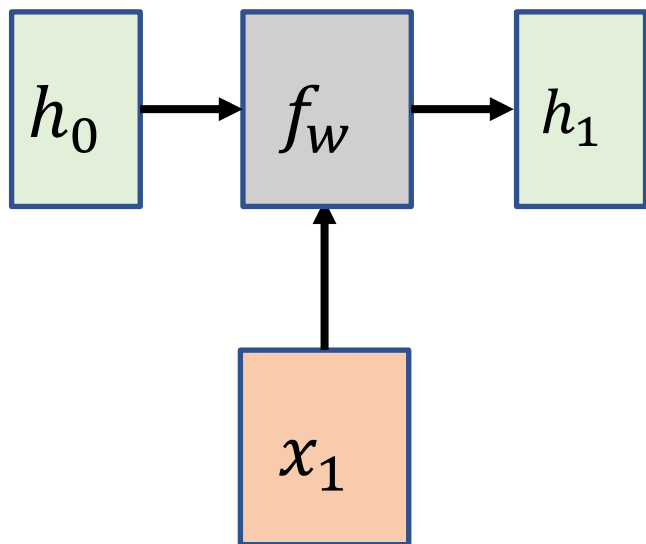
입력 시퀀스(input sequence)

음악에서 음계들의 시퀀스(일련의 사건들)

동영상의 이미지의 시퀀스(이미지의 순서에 따른 이미지)

=> 완성된 어떤 것 안에서의 일련의 순서가 있는 하나의 글자, 단어 또는 이미지 등을 말한다.

04 RNN 기본 구조의 이해(2)



$$\boxed{h_t} = \boxed{f_w} (\boxed{h_{t-1}}, \boxed{x_t})$$

new state

Parameters W
(W 로 이루어진 함수)

old state
(전상태)

input vector
(매 단계의)

04 RNN 기본 구조의 이해(2)

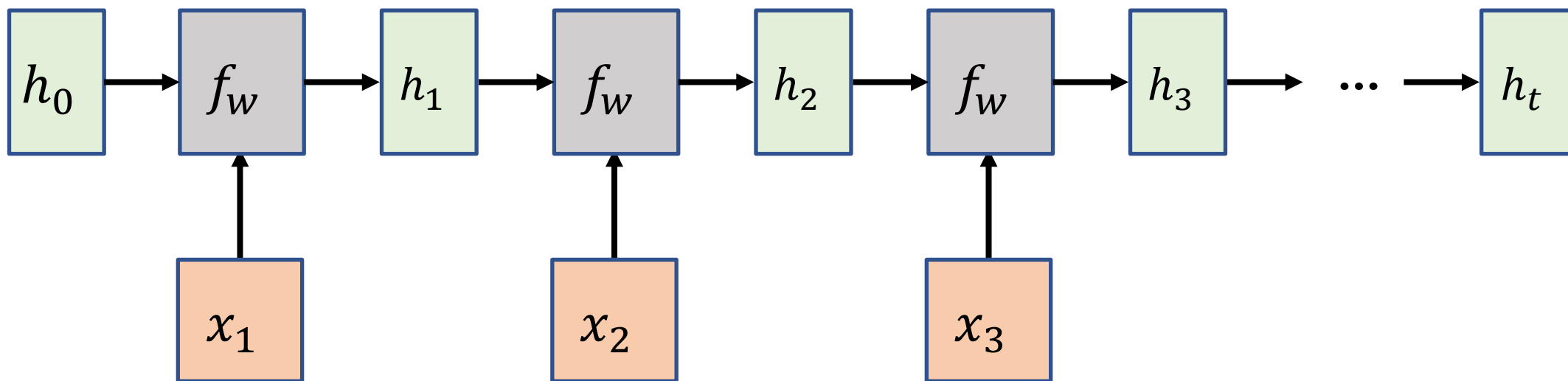
$$\boxed{h_t} = \boxed{f_w} \left(\boxed{h_{t-1}}, \boxed{x_t} \right)$$

new state

Parameters W
(W로 이루어진 함수)

old state
(전상태)

input vector
(매 단계의)



04 RNN 기본 구조의 이해(2)

$$\boxed{h_t} = \boxed{f_w}(\boxed{h_{t-1}}, \boxed{x_t})$$

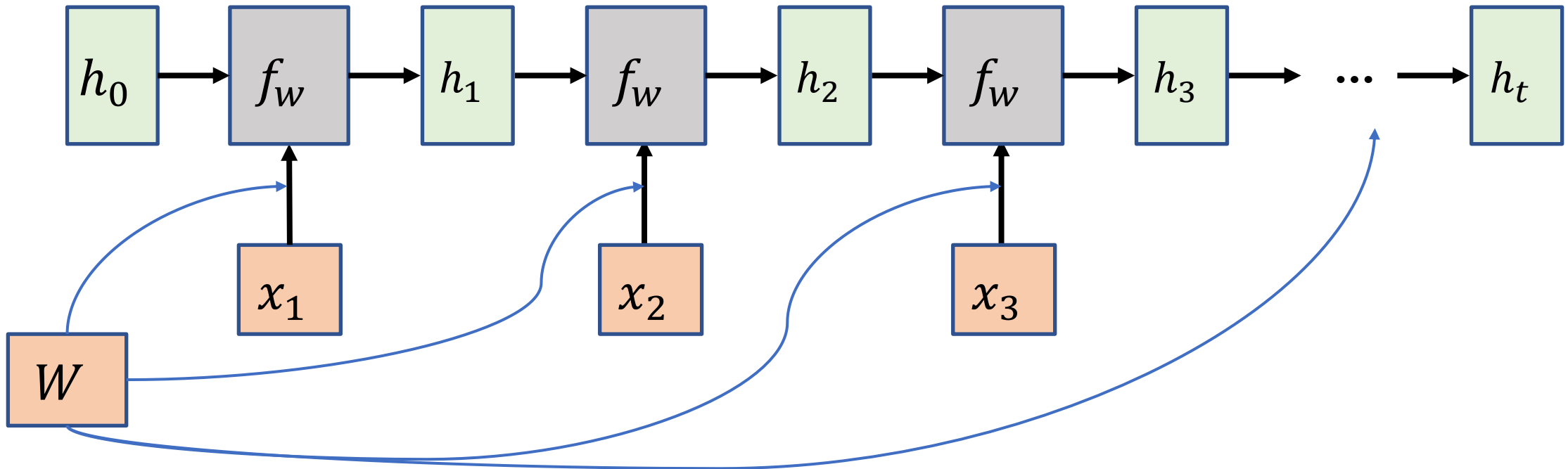
new state

Parameters W
(W 로 이루어진 함수)

old state
(전상태)

input vector
(매 단계의)

매 시간이 단계에서 W 의 가중치 값을 재사용한다.



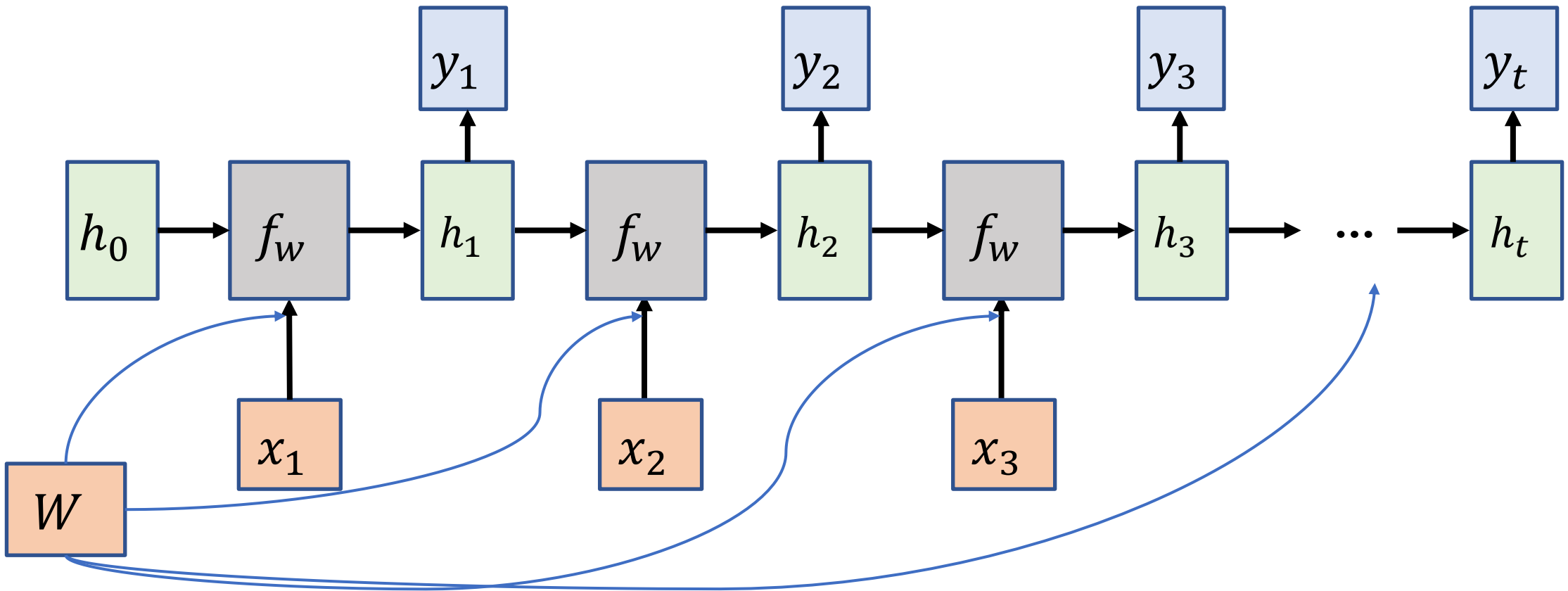
04 RNN 기본 구조의 이해(2) - Many to Many

f_w 활성화 함수로 주로 하이퍼볼릭 탄젠트 또는 ReLU와 같은 함수 사용.

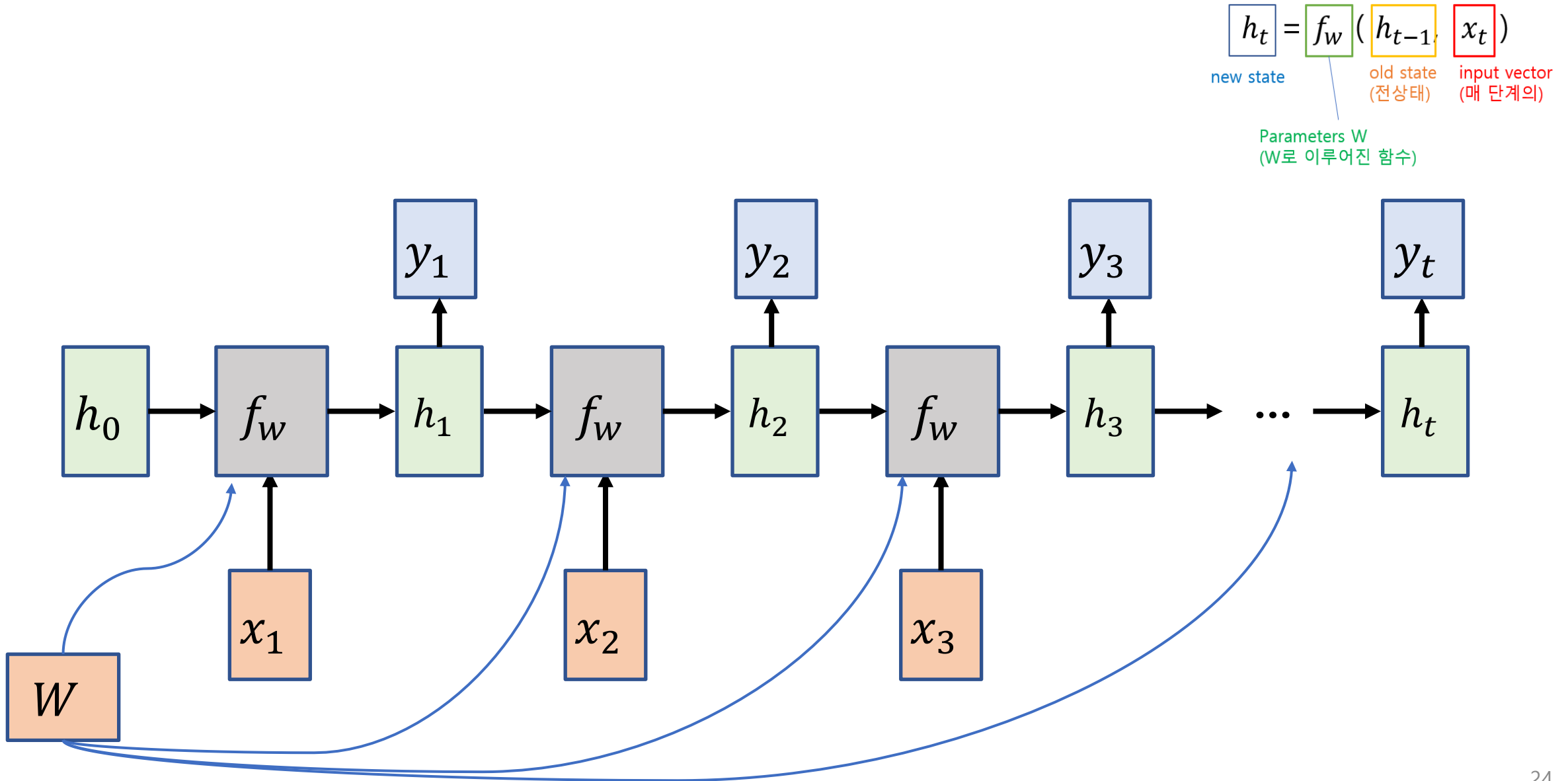
h_0 0으로 초기화 하거나 h_0 을 학습 가능한 변수로 설정하거나 등.

W W 는 학습 과정에서 최적화되는 파라미터. RNN이 학습할 때, W 는 점진적으로 업데이트 된다.

$$h_t = f(W \cdot [h_{t-1}, x_t])$$



04 RNN 기본 구조의 이해(2) - Many to Many



04 RNN 기본 구조의 이해(2) - Many to One

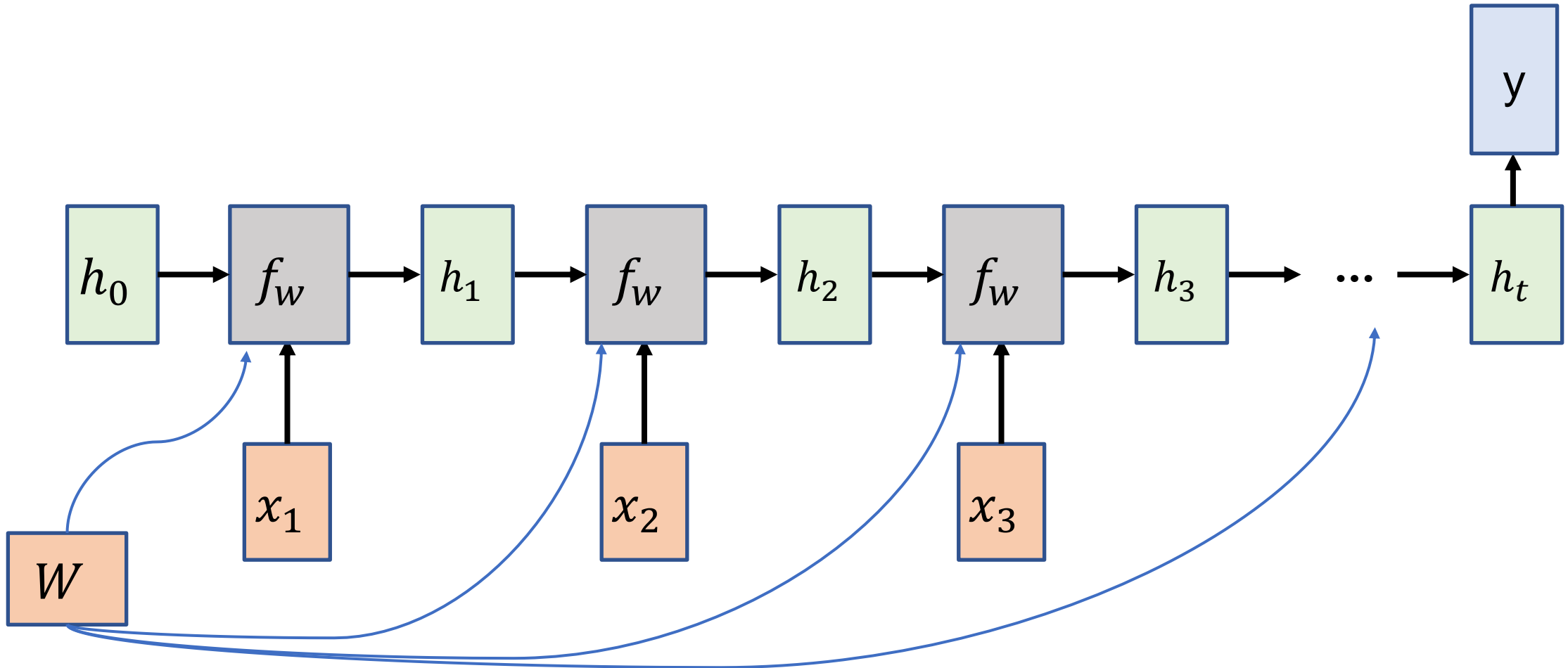
$$h_t = f_W(h_{t-1}, x_t)$$

new state

Parameters W
(W 로 이루어진 함수)

old state
(전상태)

input vector
(매 단계의)



04 RNN 기본 구조의 이해(2) - One to Many

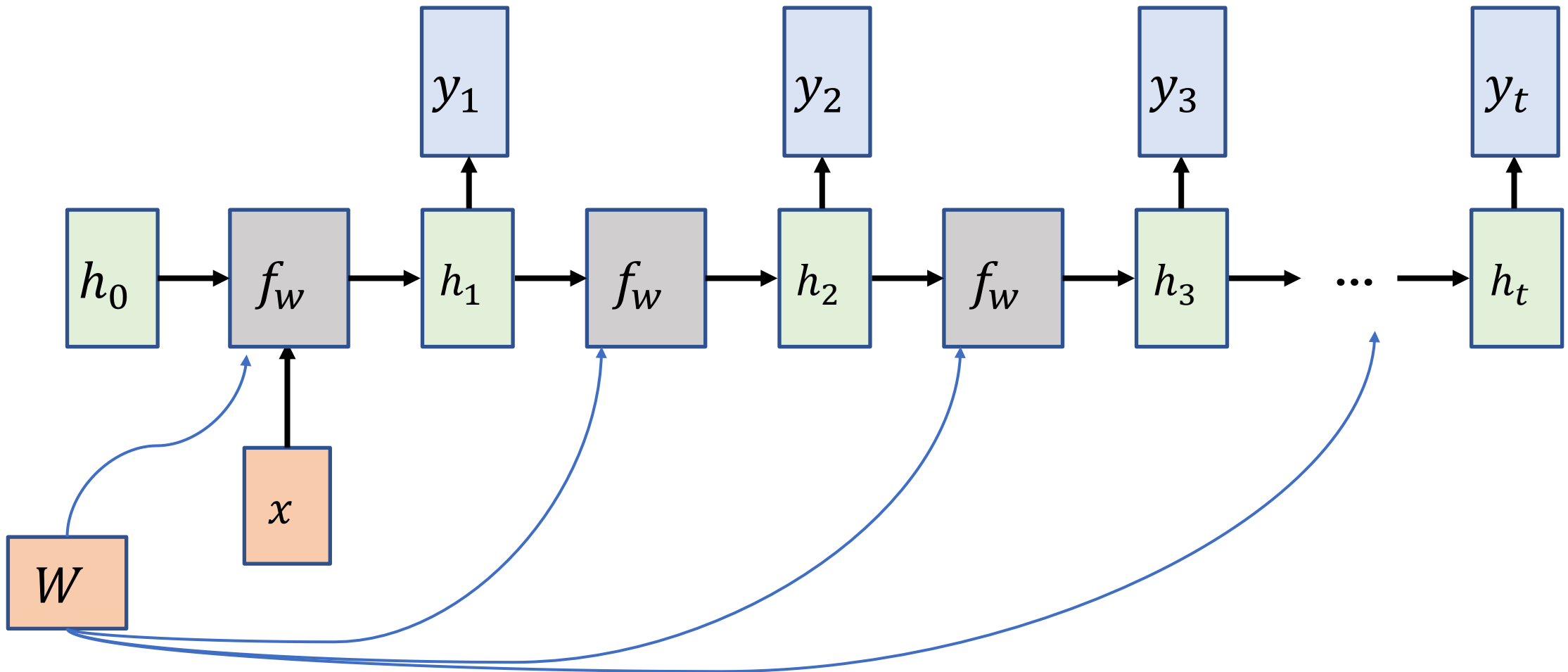
$$h_t = f_W(h_{t-1}, x_t)$$

new state

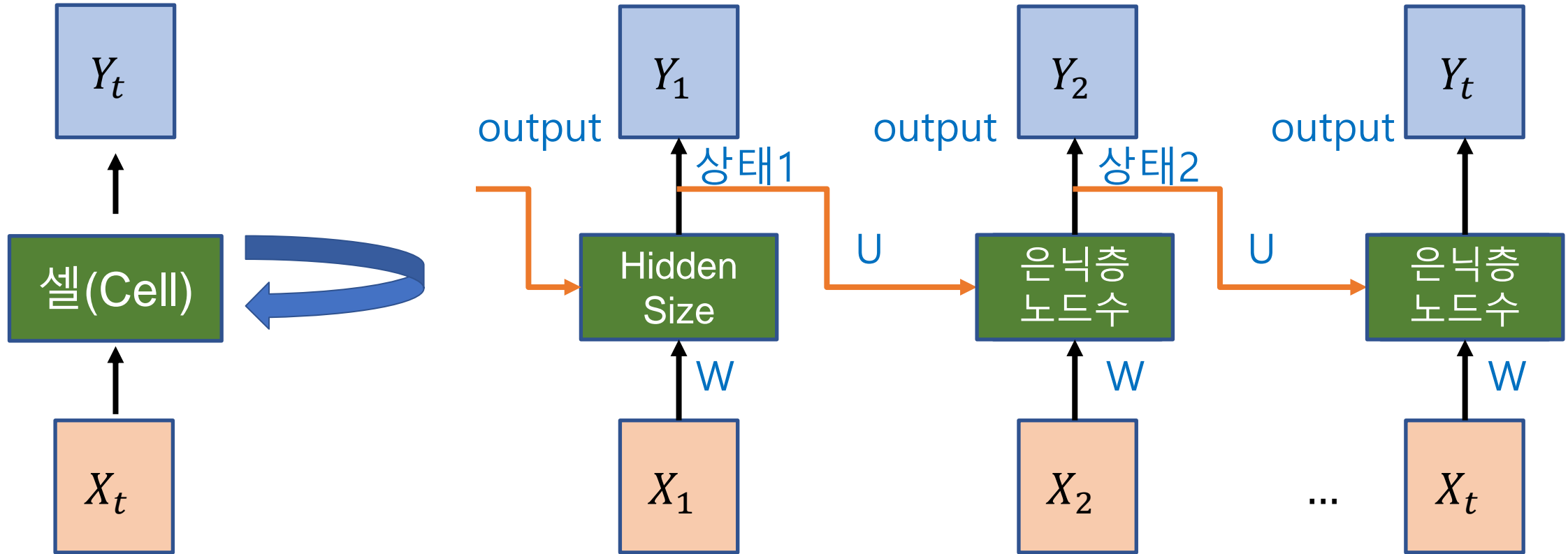
Parameters W
(W 로 이루어진 함수)

old state
(전상태)

input vector
(매 단계의)



04 RNN 기본 구조의 이해(2) - Many to Many



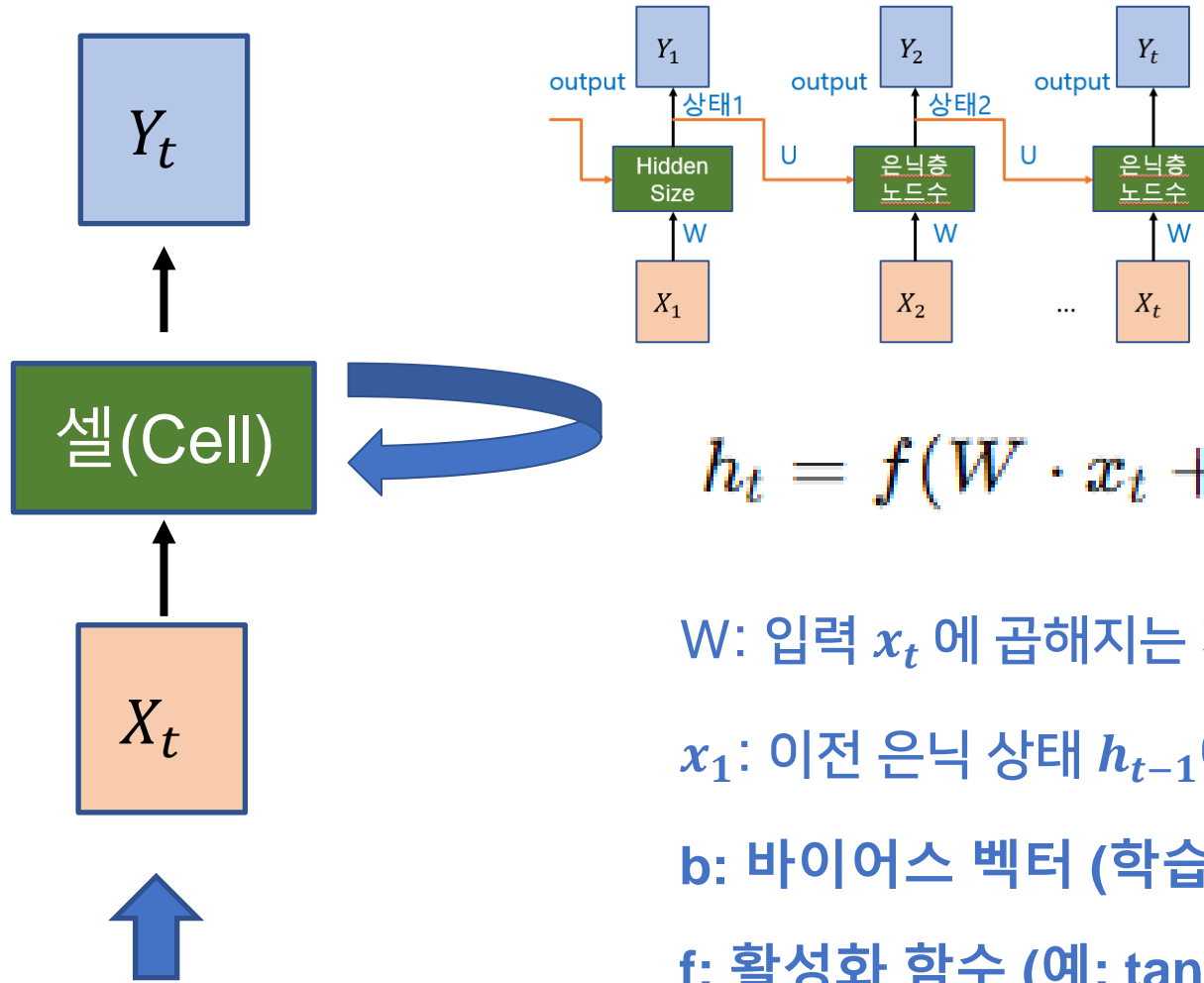
01. 각 셀을 거쳐 각각의 output가 나온다.

02. 각각의 output은 다음단계의 입력으로 사용된다.

Batch_size (100, 32)
100개의 데이터 32차원

$$\text{결과}_t = \text{activation}(W * X(t) + U * \text{상태}(t) + b)$$

04 RNN 기본 구조의 이해(2) - Many to Many



$$h_t = f(W \cdot x_t + U \cdot h_{t-1} + b)$$

W : 입력 x_t 에 곱해지는 가중치 행렬

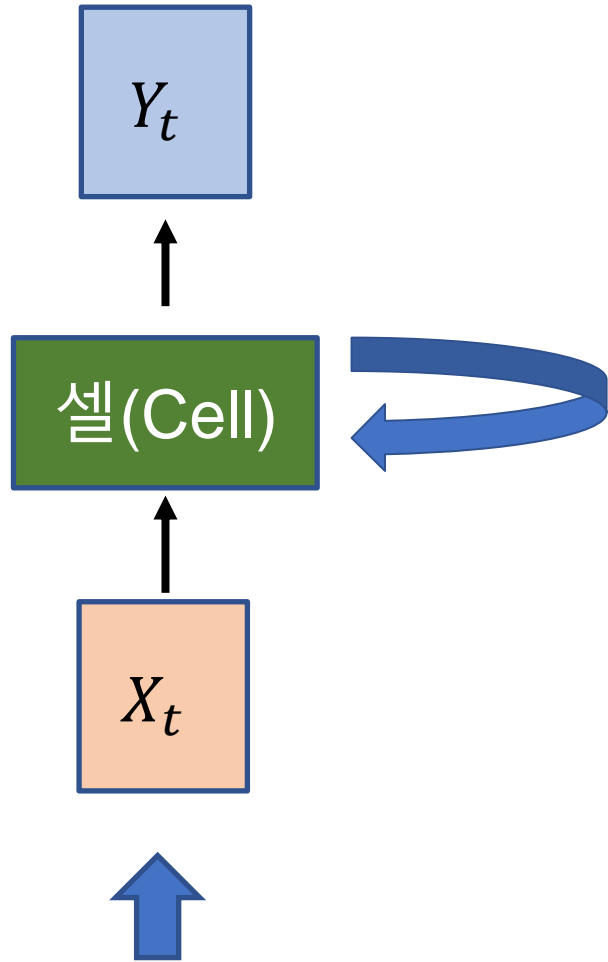
x_1 : 이전 은닉 상태 h_{t-1} 에 곱해지는 가중치 행렬

b : 바이어스 벡터 (학습 가능한 파라미터)

f : 활성화 함수 (예: tanh, ReLU)

Batch_size (100, 32)
100개의 데이터 32차원

04 RNN 기본 구조의 이해(2) - Many to Many



학습되는 가중치

W (출력 특성 차원, 입력특성차원)

U (출력 특성 차원, 출력특성차원)

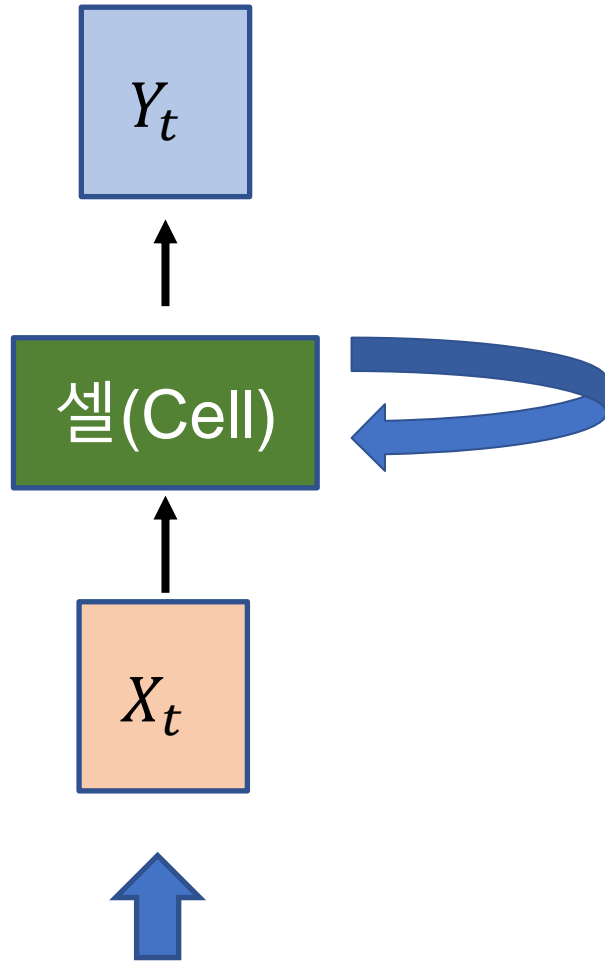
B (출력 특성 차원)

$O = \text{활성화함수}(W * X + U * h_{t-1} + b)$

최종결과 (배치 사이즈, 출력차원)

Batch_size (100, 32)
100개의 데이터 32차원

04 RNN 기본 구조의 이해(2) - Many to Many



입력 특성 차원이 32
출력 특성 차원이 64
배치 사이즈가 100

$$O = \text{활성화함수}(W * X + U * O_{t-1} + b)$$

최종결과 (100, 32)

Batch_size (100, 32)
100개의 데이터 32차원

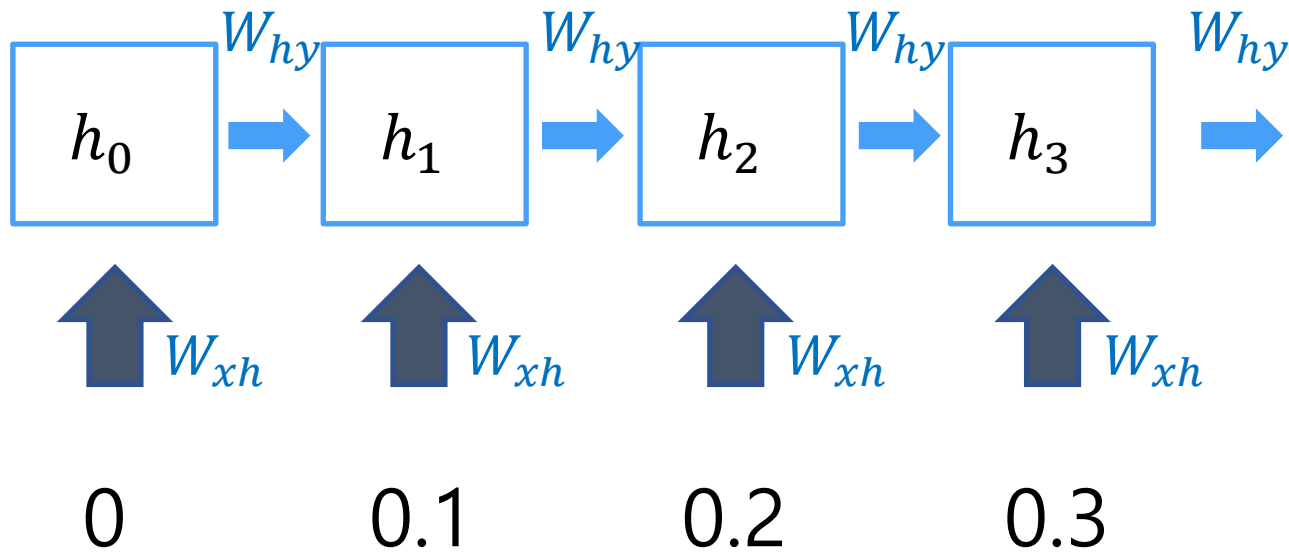
실습 : 순서가 있는 숫자 데이터 마지막 값 예측

05 [실습] 숫자 데이터 예측

출력(output)

y_3 0.4

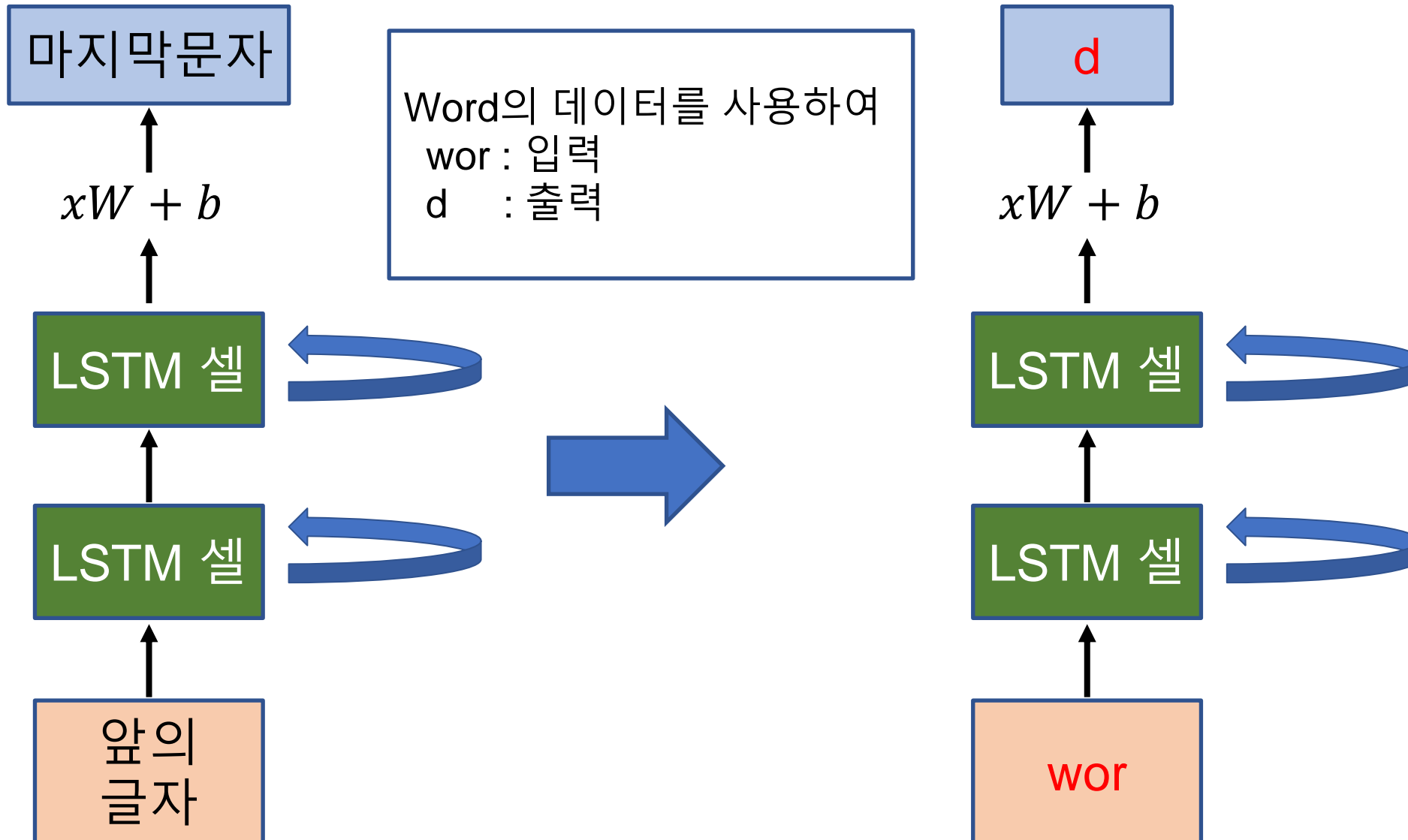
최종 평가 : mse



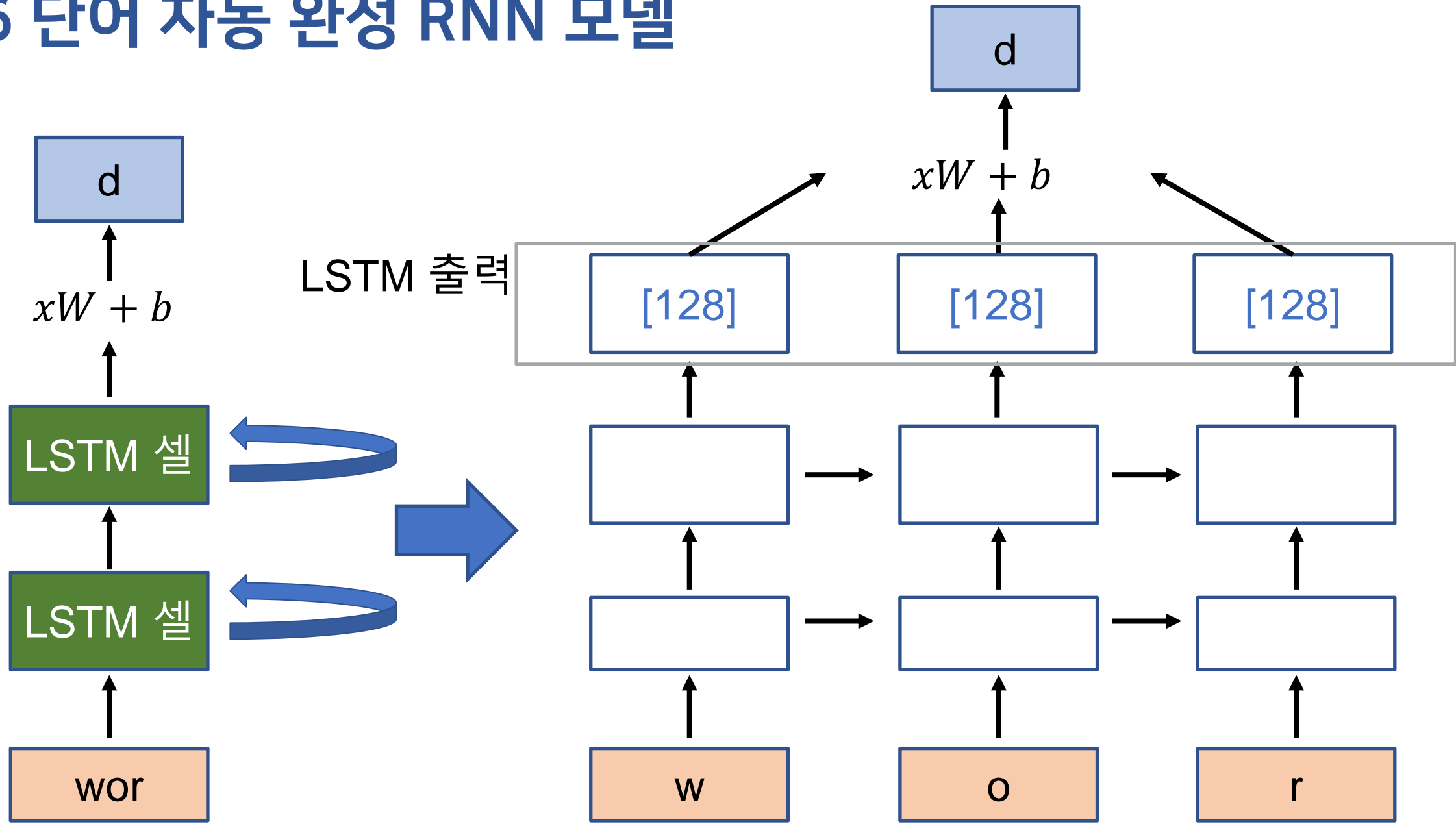
입력(input)

실습 : 단어 자동 완성을 RNN 모델을 활용해 보자.

06 단어 자동 완성 RNN 모델



06 단어 자동 완성 RNN 모델



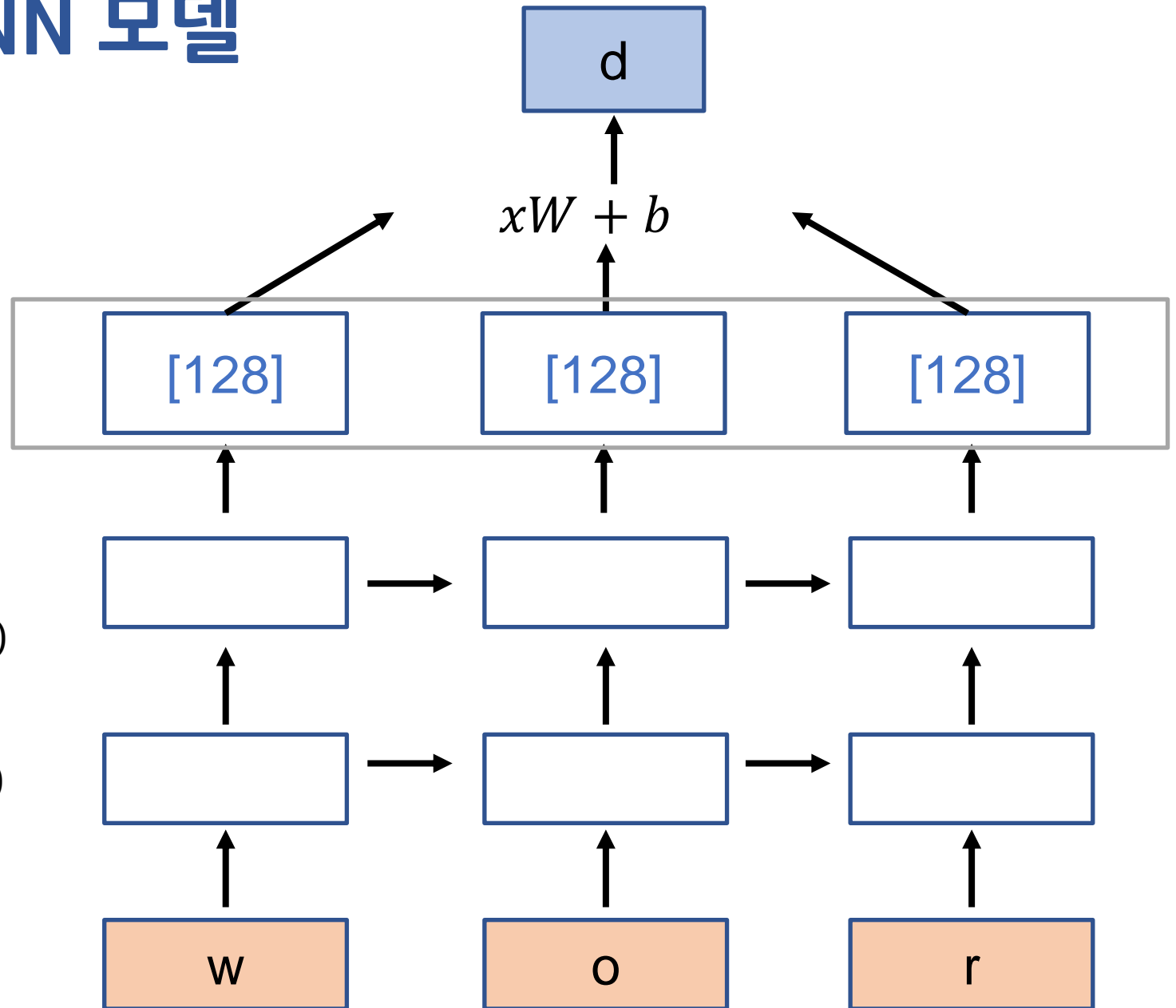
06 단어 자동 완성 RNN 모델

입력은 ONE-HOT-ENCODING 형식으로

영문자 W의 경우 (W에 해당되는 위치가 1로 표시)
[0, 0, 0, 0, ..., 1, 0, 0, 0]

영문자 B의 경우 (W에 해당되는 위치가 1로 표시)
[0, 1, 0, 0, ..., 0, 0, 0, 0]

LSTM 출력



06 단어 자동 완성 RNN 모델

(1) 학습할 데이터 셋

- A. 고정 단어길이의 단어들
- B. 입력과 출력으로 나누기

(2) 모델 구성

- A. LSTM 셀(Cell1, Cell2), MultiCell
- B. `Dynamic_rnn` 함수 사용 -> 심층 순환 신경망

(3) 데이터 전 처리

- A. 입력과 출력값 비교를 위한 영단어 글자의 원 핫 인코딩

(4) 학습 및 예측

- A. 손실함수로 0,1로 이루어진 벡터가 이니기에 `sparse_softmax_cross_entropy_with_logits` 를 사용가능

(5) 평가

07 구글 번역 모델

- 2016년 - 기존의 통계 기반 기계 번역에서 신경망 기반 기계 번역으로 전환.

구글의 기계 번역에 사용한 Sequence to Sequence 모델(Seq2Seq)

- 2017년 이후 - Transformer 기반 신경망 기반 기계 번역

- 현재 대부분 Transformer 기반 모델 사용. 최근에는 BERT, GPT같은 프리트레이닝 언어 모델들이 자연어 처리(NLP)작업에서 활용.

08 LSTM(Long Short Term Memory)

왜? LSTM은 나왔을까?

- 앞의 내용이 소실되는 문제를 해결.
- LSTM이 나온 이후에 성능 개선이 많이 일어났다.

08 LSTM(Long Short Term Memory)

- (1) 컴퓨터가 단어나 문장을 이해하고 예측할 때 사용되는 특별한 종류의 인공 신경망
- (2) 1997년: Sepp Hochreiter와 Jürgen Schmidhuber가 처음 제안한 모델
- (3) LSTM은 기본적으로 과거의 정보를 기억하고 이를 다음 예측에 반영하는데 강점을 지닌다.
- (4) 자연어 처리(NLP)분야에서 많이 사용되었던 모델 중 하나. 최근은 Transformer을 사용한다.

09 Transformer 모델

- (1) Transformer는 RNN(Recurrent Neural Network)와는 다른 구조를 가진 모델
- (2) NLP 작업에서는 RNN 기반 모델들을 거의 대체하고 있다.