

# 딥러닝 입문 (RNN)

# 01 RNN은 약자의 의미

**Recurrent Neural Networks(RNN) : 순환 신경망이라 말한다.**

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

## 02 RNN은 어디에 쓰일까?

### (1) 기계 번역(Neural Machine Translation) - NMT

#### A. 하나의 예로 구글 뉴럴 기계 번역기(2016년도)

- 기존의 성능을 크게 뛰어 넘음
- 몇몇 언어에서는 인간에 가까운 수준까지 도달

## 02 RNN은 어디에 쓰일까?

### (1) 기계 번역(Neural Machine Translation) - NMT

- 전체 구성은 Encoder-Decoder로 이루어짐.
- Encoder와 Decoder는 각각 8층의 순환신경망으로 구성됨.  
(순환 신경망의 기본 모듈로는 LSTM을 사용)

## 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.

## 02 RNN은 어디에 쓰일까?

### (3) 음성 인식기

A. 하나의 딥러닝 모델로 영어와 중국어를 모두 인식함.

## 02 RNN은 어디에 쓰일까?

(5) 딥러닝 모델을 조합하여 이미지의 캡션을 생성.

A. CNN과 RNN을 조합하여 이미지의 캡션을 자동으로 생성한다.



참조

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

## 02 RNN은 어디에 쓰일까?

### (6) 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

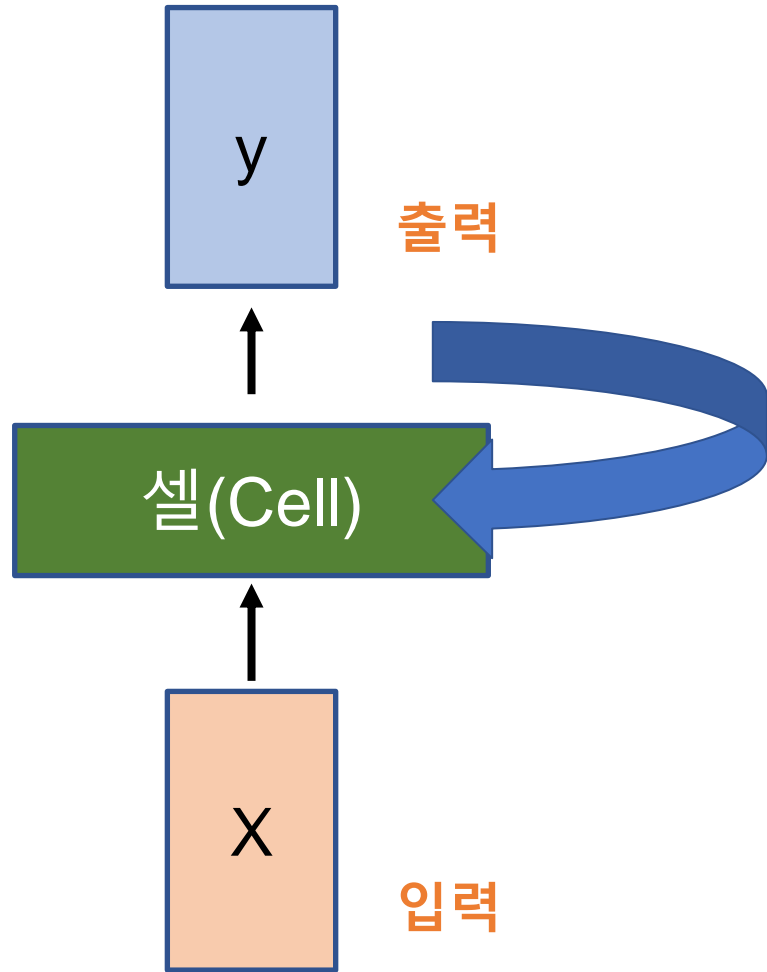


## 02 RNN은 어디에 쓰일까?

### (7) 주가 예측 - 예측 분석

#### A. 주식 예측

## 03 RNN의 기본 구조 및 이해

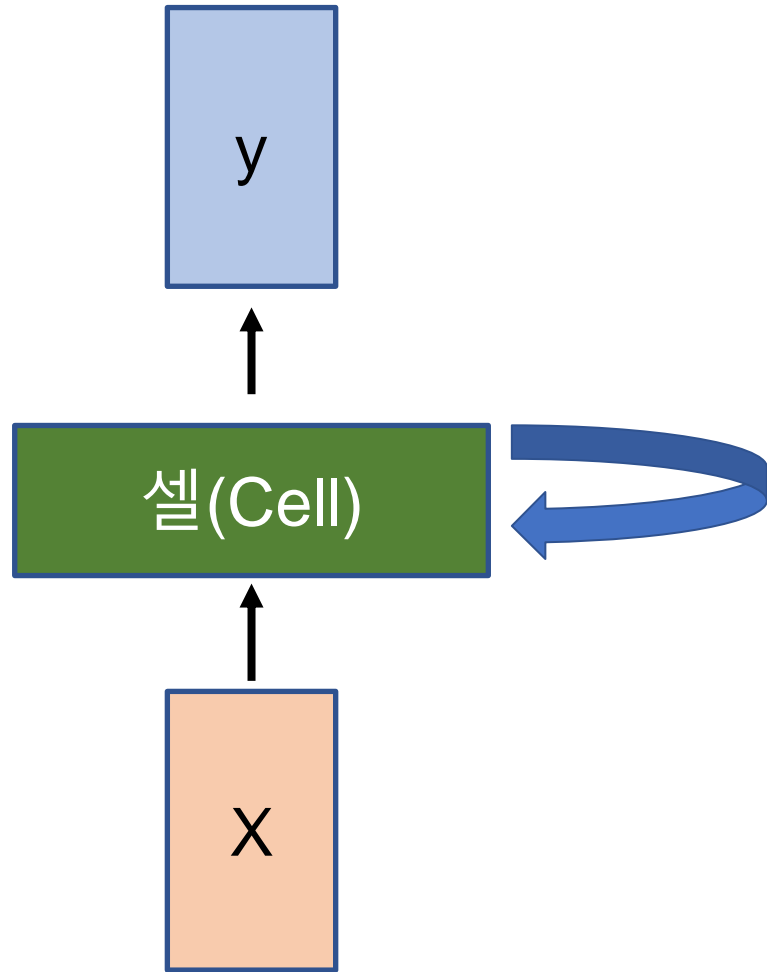


출력

순환 연결

- 하나의 셀의 결과가 다음 셀의 입력의 하나로 들어가 적용된다.

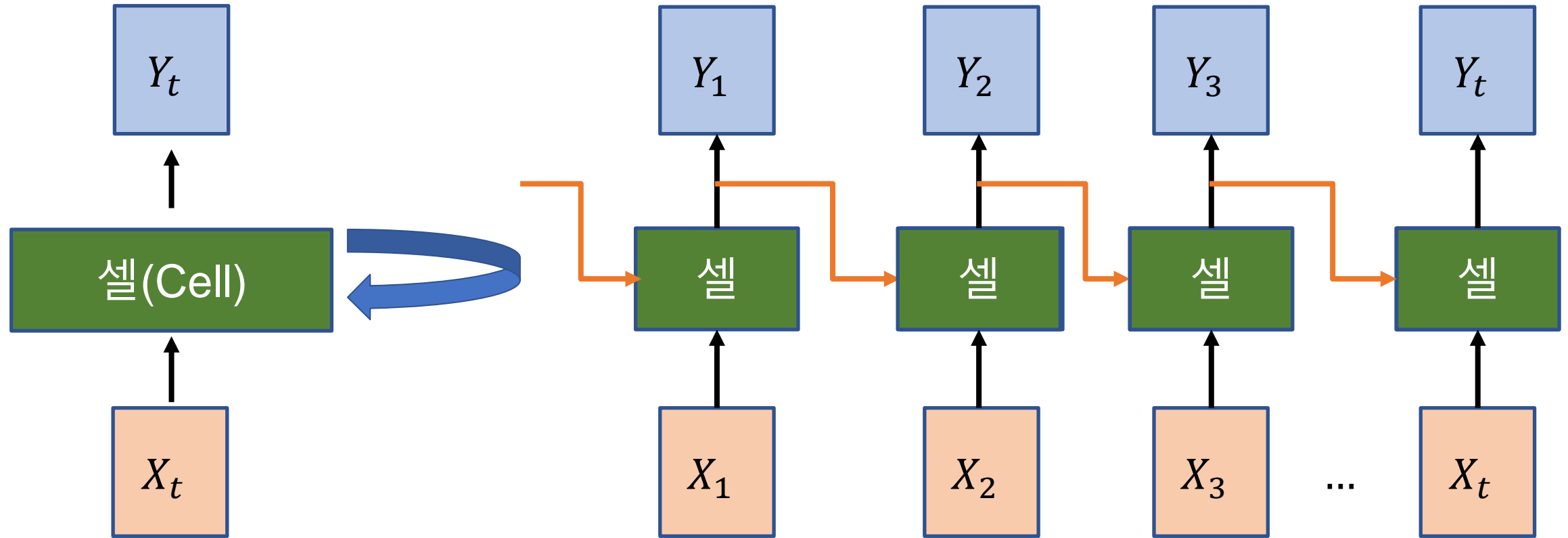
## 03 RNN의 기본 구조 및 이해



### \* Key point

- RNN은 셀을 여러 개 중첩하여 심층 신경망을 만든다.

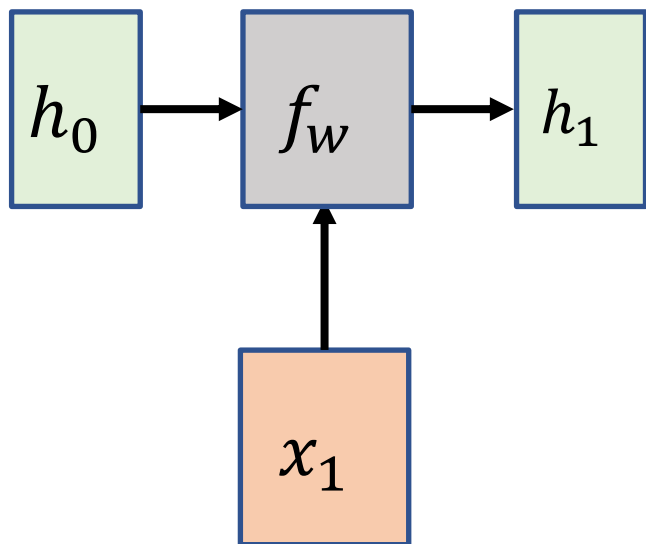
## 03 RNN의 기본 구조 및 이해



### \* Key point

- 앞단계에서 학습한 결과를 다음 단계의 학습에 이용.
- 학습 데이터를 단계별로 구분하여 입력을 하여야 함.  
( $X_1, X_2, X_3, \dots, X_t$ )

### 03 RNN: 계산 그래프



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

new state

Parameters  $W$   
( $W$ 로 이루어진 함수)

old state  
(전상태)

input vector  
(매 단계의)

# 03 RNN: 계산 그래프

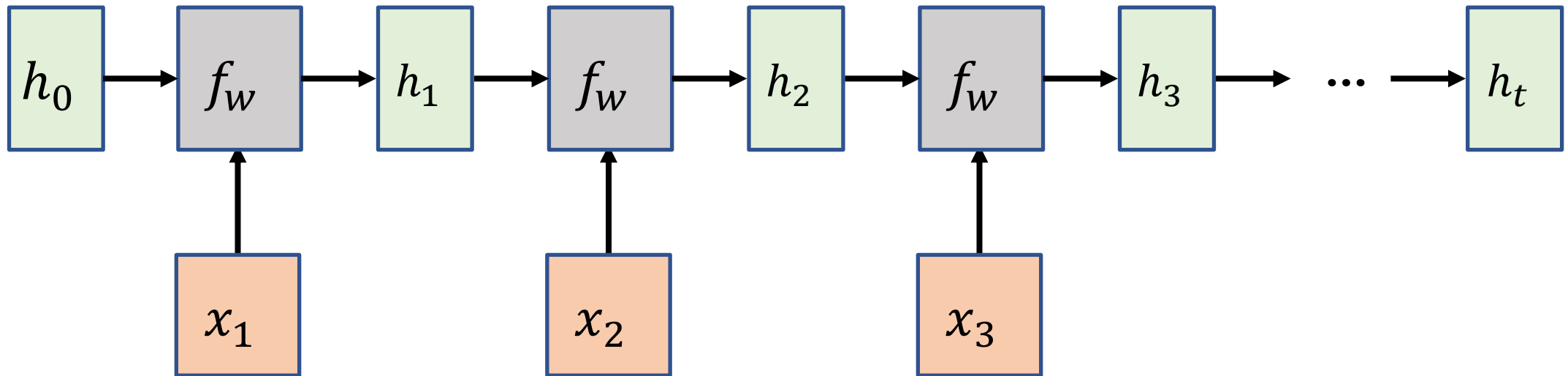
$$\boxed{h_t} = \boxed{f_w} \left( \boxed{h_{t-1}}, \boxed{x_t} \right)$$

new state

Parameters W  
(W로 이루어진 함수)

old state  
(전상태)

input vector  
(매 단계의)



## 03 RNN: 계산 그래프

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

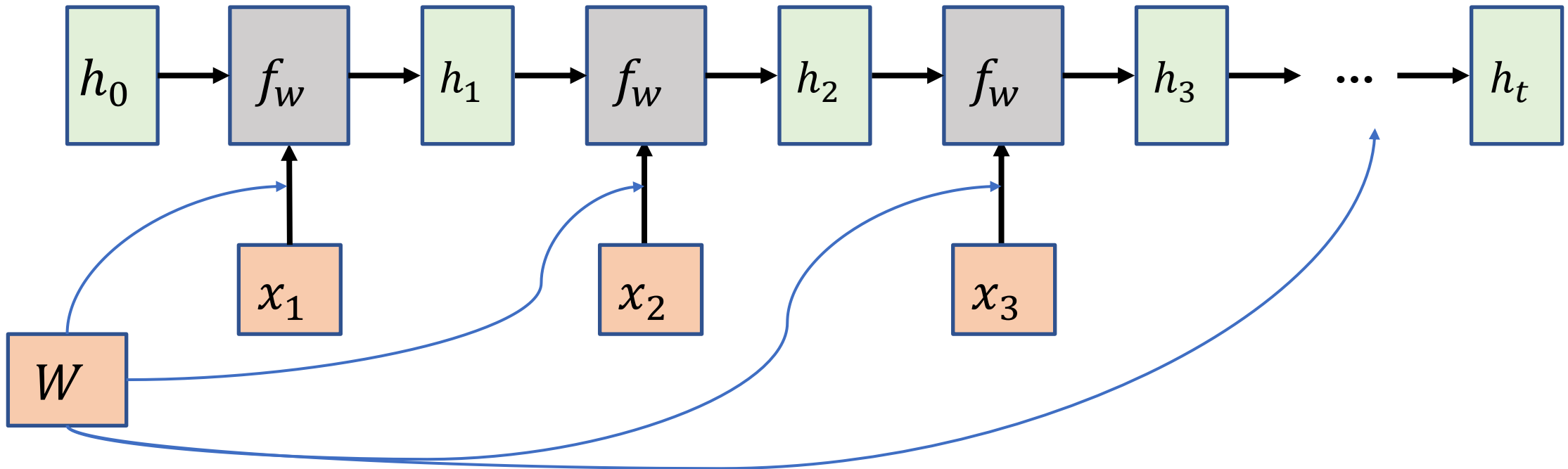
new state

Parameters  $W$   
( $W$ 로 이루어진 함수)

old state  
(전상태)

input vector  
(매 단계의)

매 시간이 단계에서  $W$ 의 가중치 값을 재사용한다.



# 03 RNN: 계산 그래프 : Many to Many

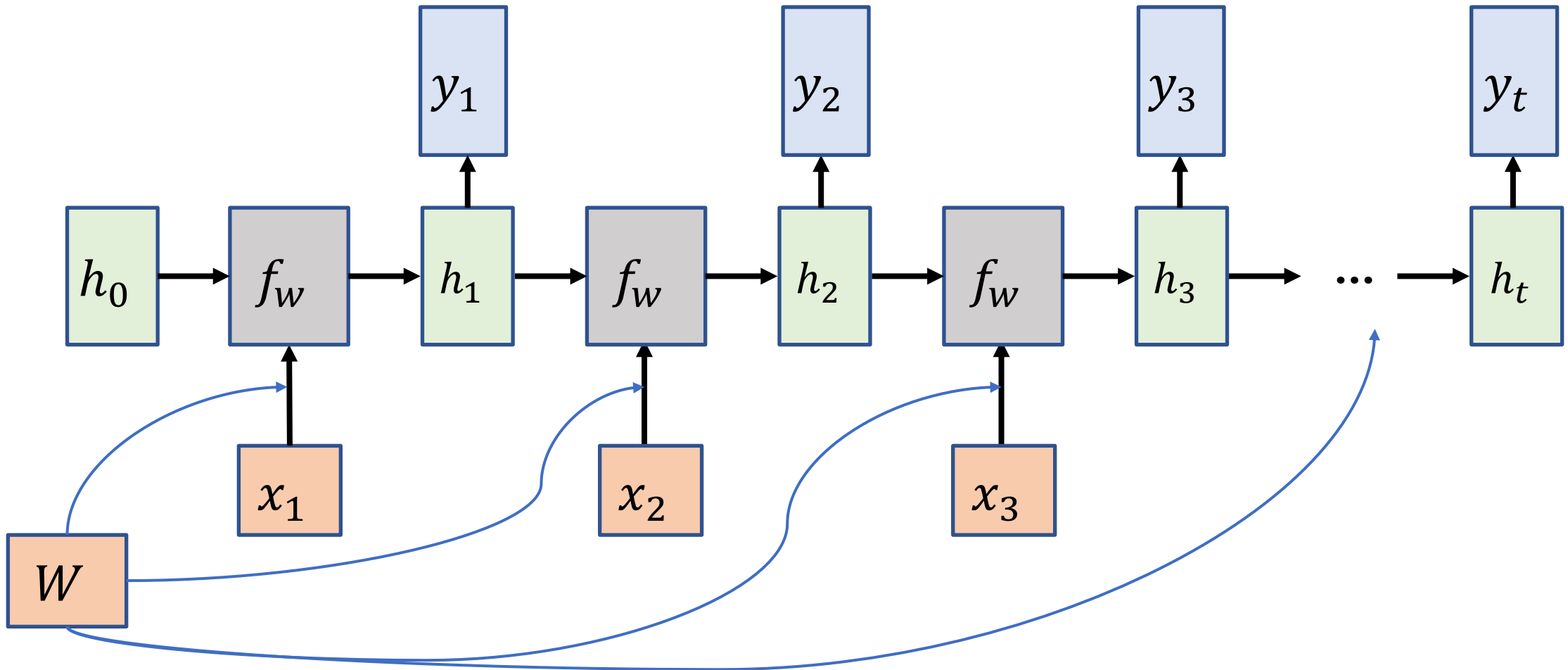
$$h_t = f_w(h_{t-1}, x_t)$$

new state

Parameters W  
(W로 이루어진 함수)

old state  
(전상태)

input vector  
(매 단계의)





# 03 RNN: 계산 그래프 : Many to Many

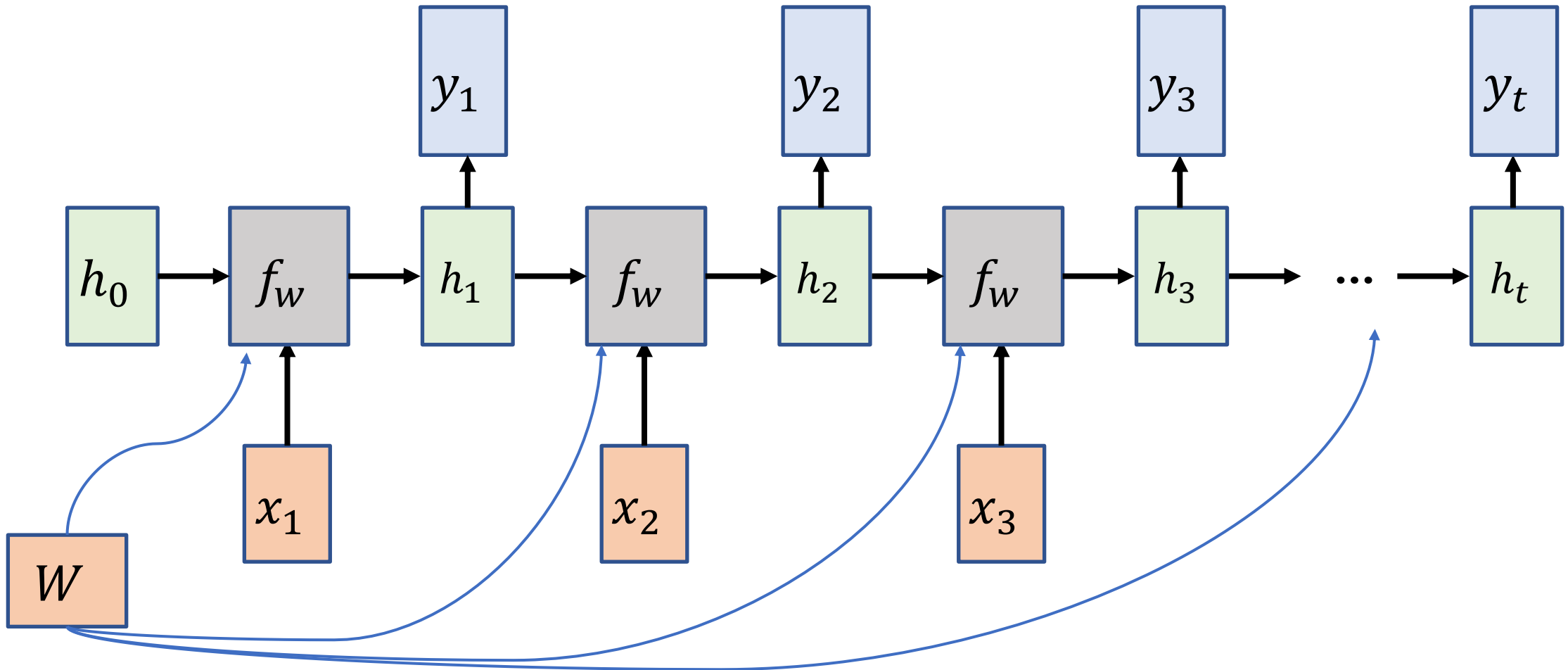
$$h_t = f_W(h_{t-1}, x_t)$$

new state

Parameters  $W$   
( $W$ 로 이루어진 함수)

old state  
(전상태)

input vector  
(매 단계의)



# 03 RNN: 계산 그래프 : Many to One

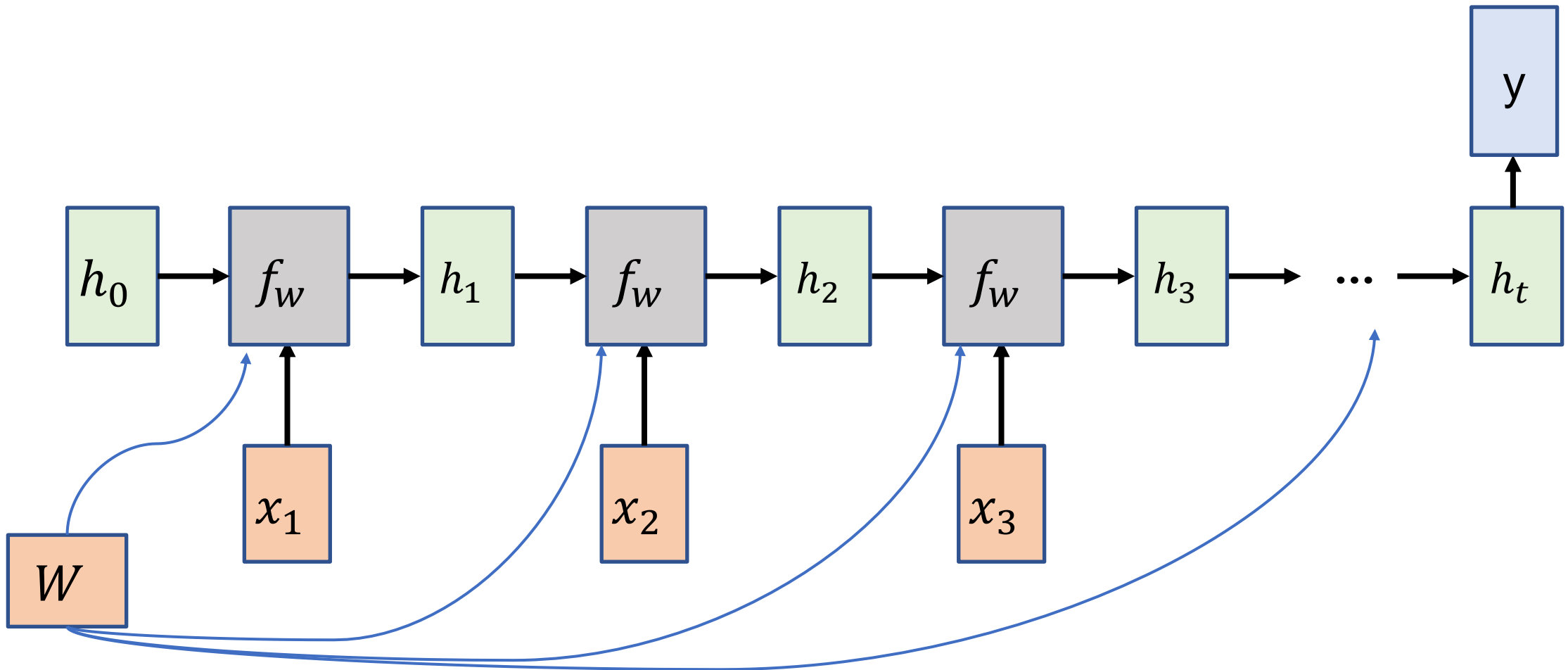
$$h_t = f_W(h_{t-1}, x_t)$$

new state

Parameters W  
(W로 이루어진 함수)

old state  
(전상태)

input vector  
(매 단계의)



# 03 RNN: 계산 그래프 : One to Many

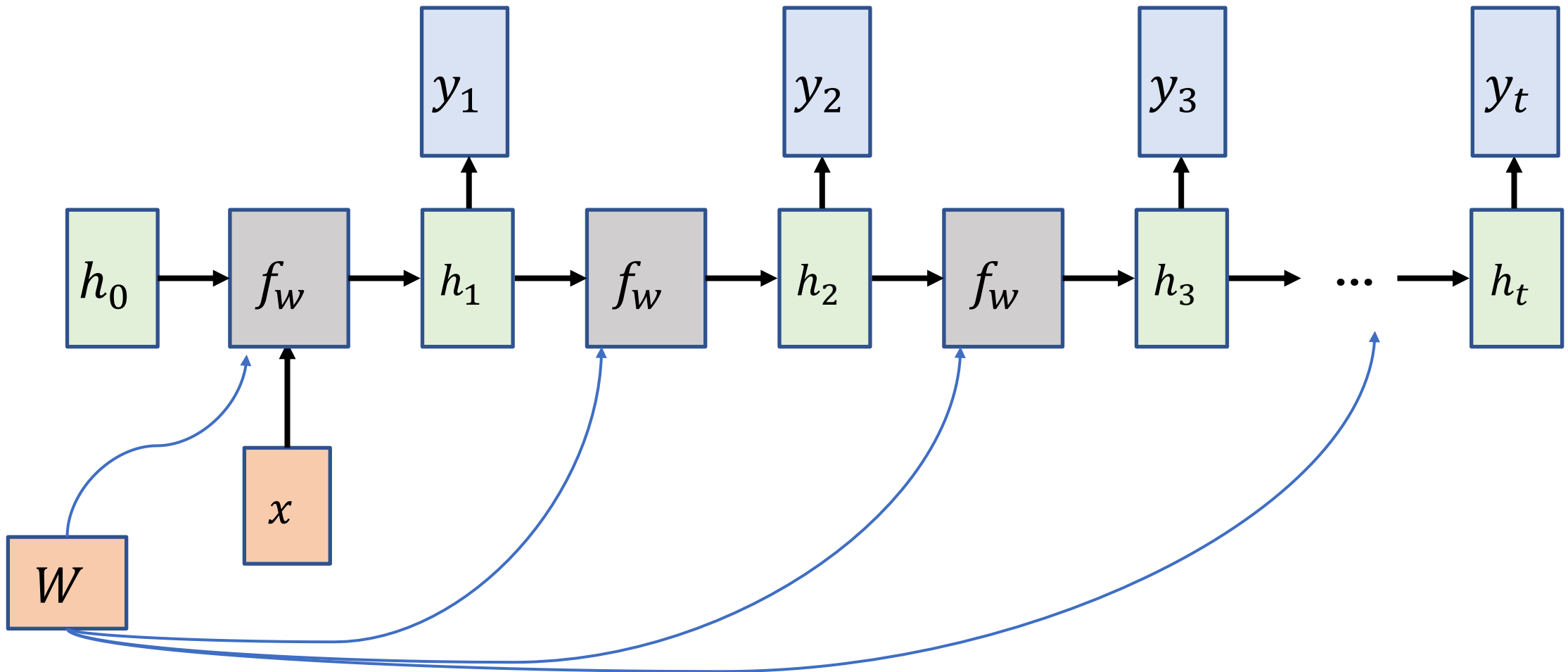
$$h_t = f_W(h_{t-1}, x_t)$$

new state

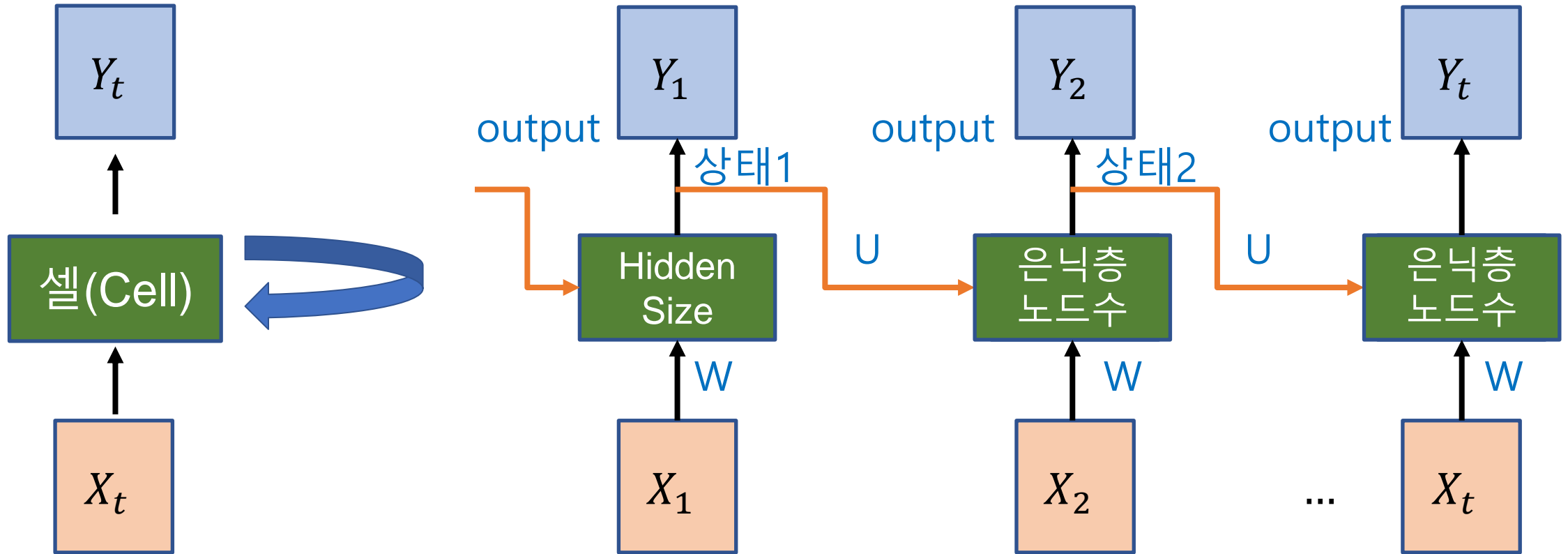
old state (전상태)

input vector (매 단계의)

Parameters W (W로 이루어진 함수)



## 04 RNN의 기본 구조 및 이해 – Many to Many



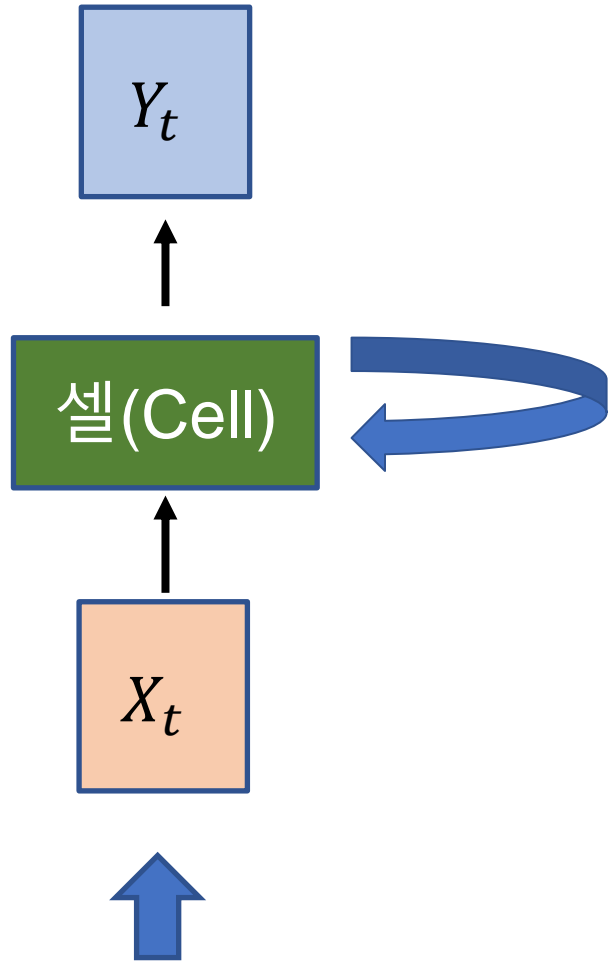
01. 각 셀을 거쳐 각각의 output가 나온다.

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 RNN의 기본 구조 및 이해 – Many to Many



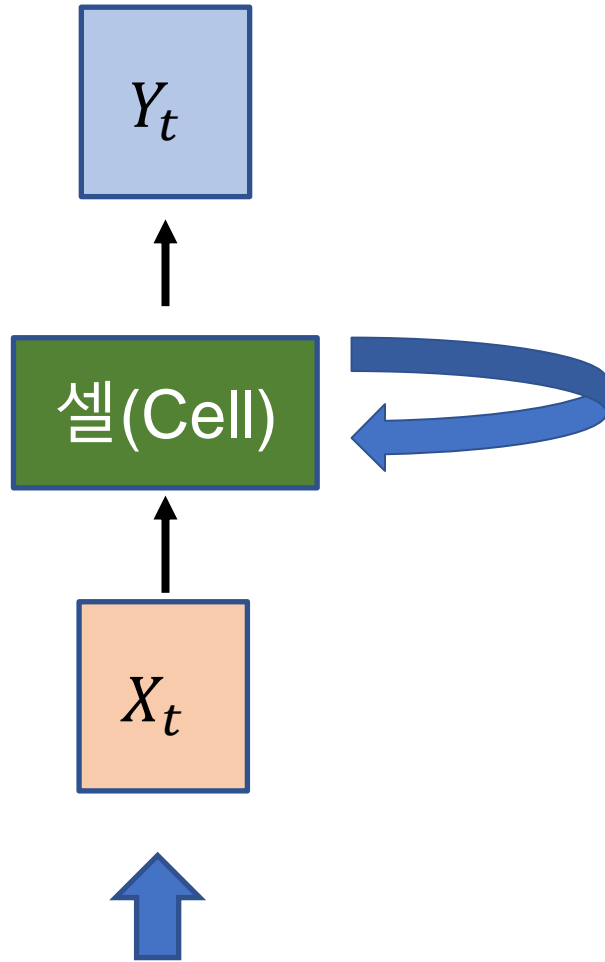
$W$  (출력 특성 차원, 입력특성차원)  
 $U$  (출력 특성 차원, 출력특성차원)  
 $B$  (출력 특성 차원)

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

최종결과 (배치 사이즈, 출력차원)

Batch\_size (100, 32)  
100개의 데이터 32차원

## 04 RNN의 기본 구조 및 이해 – Many to Many



입력 특성 차원이 32  
출력 특성 차원이 64  
배치 사이즈가 100

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

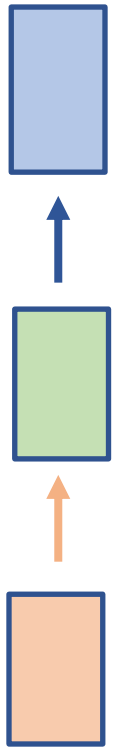
최종결과 (100, 32)

Batch\_size (100, 32)  
100개의 데이터 32차원

# RNN의 다양한 구조

## 05 다양한 형태의 순환 신경망

one to one



RNN

one to many

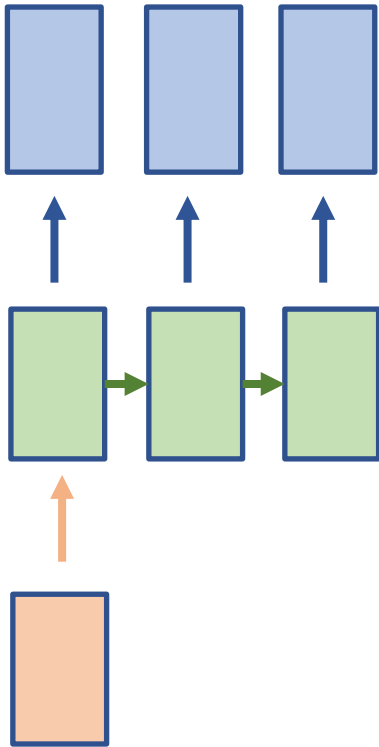
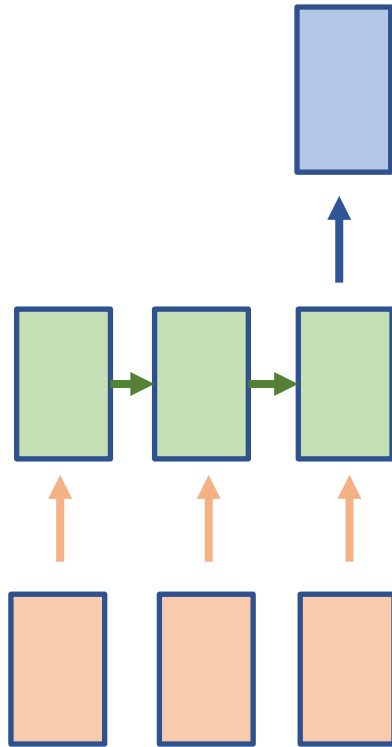


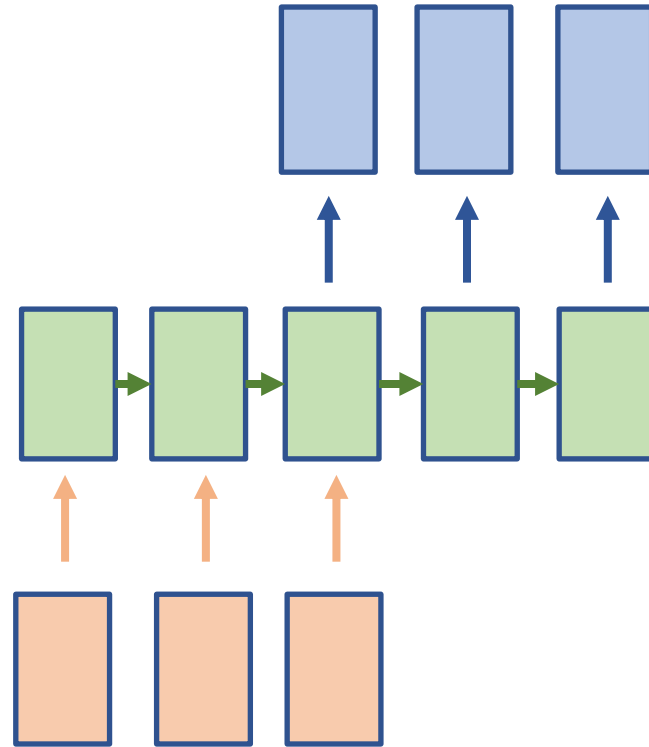
Image  
Captioning

many to one



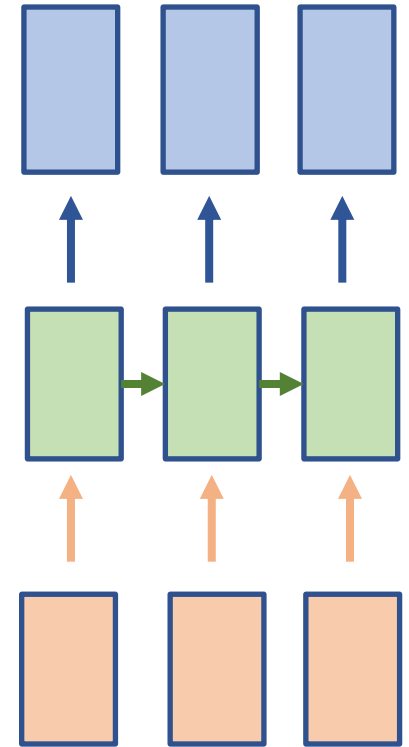
Sentiment  
Classification

many to many



Machine Translation

many to many



Video  
Classification