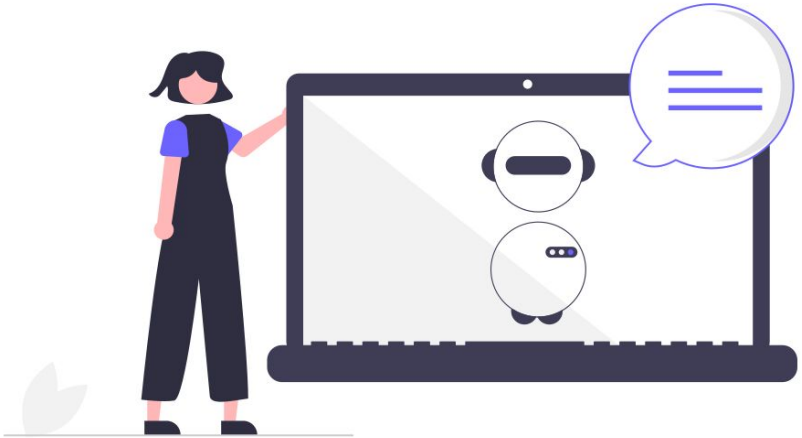
딥러닝 기초

Recurrent Neural Networks



오늘의 목차

Data Diving

- 딥러닝으로 문제를 푼다는 것은?
  - 딥러닝으로의 문제 풀이 → 뭔가를 벡터로 바꾸는 것
- text using deep learning
  - word2vec: word → vectors
  - sentence, paragraph, etc,.. → vector
- Recurrent Neural Networks

딥러닝을 이용한 문제풀이

- 딥러닝으로 문제를 푼다는 것은?
  - 입력/출력 정의
  - 입력/출력을 어떻게든 벡터로 바꾸는 것
  - 그 다음에 뉴럴넷에 그냥 학습시킴!
- 주로 하는 입력
  - 이미지
  - 텍스트
  - 표
  - 오디오
  - 그래프 (데이터구조)

이미지 → 벡터

Data Diving

- 이미지:(255 X 255 X 255짜리 1픽셀) X 화소 수
  - ex) 1920 * 1040짜리 → 3 * 1920 * 1040 텐서
- 비디오: 이미지 X 타임프레임 수

텍스트 → 벡터

Data Diving

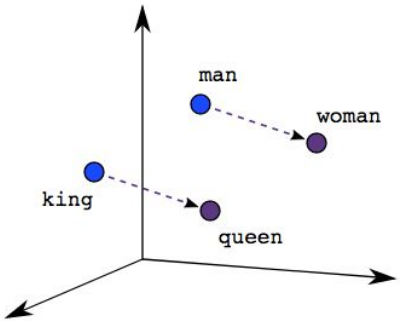
- 단어 → 벡터
  - one-hot encoding
  - 이후 잘 벡터로 만들기 (by word2vec)
- 단어는 어떻게 정의하는가? → Tokenizer
  - 어릴 때 배운 형태소분석기
  - 띄어쓰기 단위로 하기 → 영어같은 거에서는 잘 됨
  - bpe 등등 이건 따로 하나의 topic
- 데이터 안에서의 단어들: Vocabulary
  - 내가 이때까지 못 본 단어를 OOV라고 함

이미지 / 텍스트 같이 → 벡터

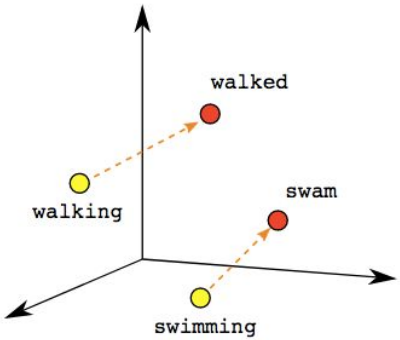
- 어차피 다 벡터인데 못 합치나?
  - multi-modality
  - GPT는 합쳐서 볼 수 있음.

AN IMAGE IS WORTH 16X16 WORDS:
TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

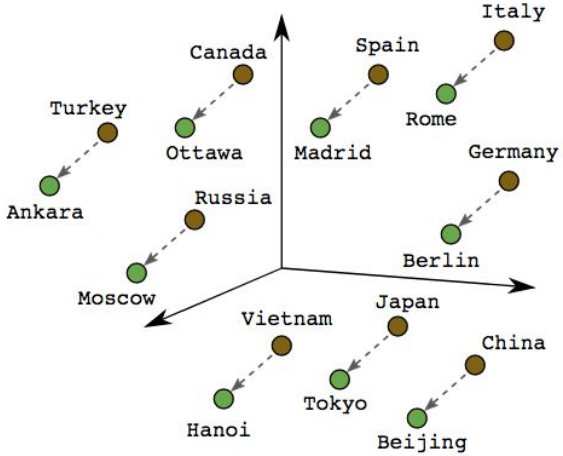
Alexey Dosovitskiy*,†, Lucas Beyer*, Alexander Kolesnikov*, Dirk Weissenborn*,
Xiaohua Zhai*, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer,
Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby*,†
*equal technical contribution, †equal advising
Google Research, Brain Team
{adosovitskiy, neilhoulsby}@google.com



Male-Female



Verb Tense



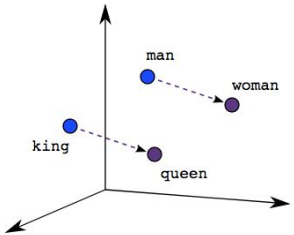
Country-Capital

단어 하나를 공간의 한 점으로 본다면...

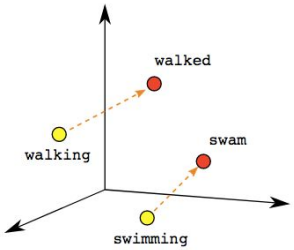
Data Diving

단어 하나를 공간의 한 점으로 본다면...

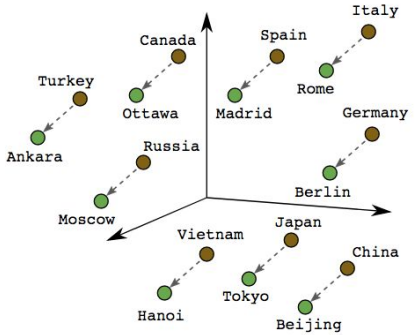
seoul = (0.1, 0.3, 0.5)
busan = (0.2, 0.5, 0.3)
tokyo = (0.1, -0.2, -0.4)



Male-Female



Verb Tense



Country-Capital

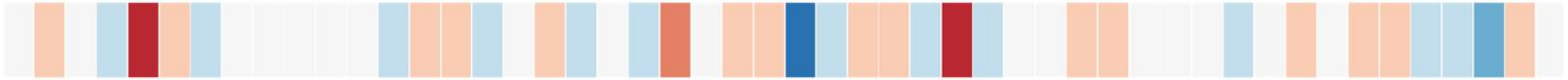
근데 사실은 3차원 공간이 아니라 n차원 공간이었다

```
seoul = [ 0.50451 , 0.68607 , -0.59517 , -0.022801, 0.60046 ,
-0.13498 , -0.08813 , 0.47377 , -0.61798 , -0.31012 , -0.076666,
1.493 , -0.034189, -0.98173 , 0.68229 , 0.81722 , -0.51874 ,
-0.31503 , -0.55809 , 0.66421 , 0.1961 , -0.13495 , -0.11476 ,
-0.30344 , 0.41177 , -2.223 , -1.0756 , -1.0783 , -0.34354 ,
0.33505 , 1.9927 , -0.04234 , -0.64319 , 0.71125 , 0.49159 ,
0.16754 , 0.34344 , -0.25663 , -0.8523 , 0.1661 , 0.40102 ,
1.1685 , -1.0137 , -0.21585 , -0.15155 , 0.78321 , -0.91241 ,
-1.6106 , -0.64426 , -0.51042 ]
```

단어 하나를 n차원의 벡터로 표현 가능
word vector 또는 word embedding이라고 표현

“king”

“Man”



“Woman”

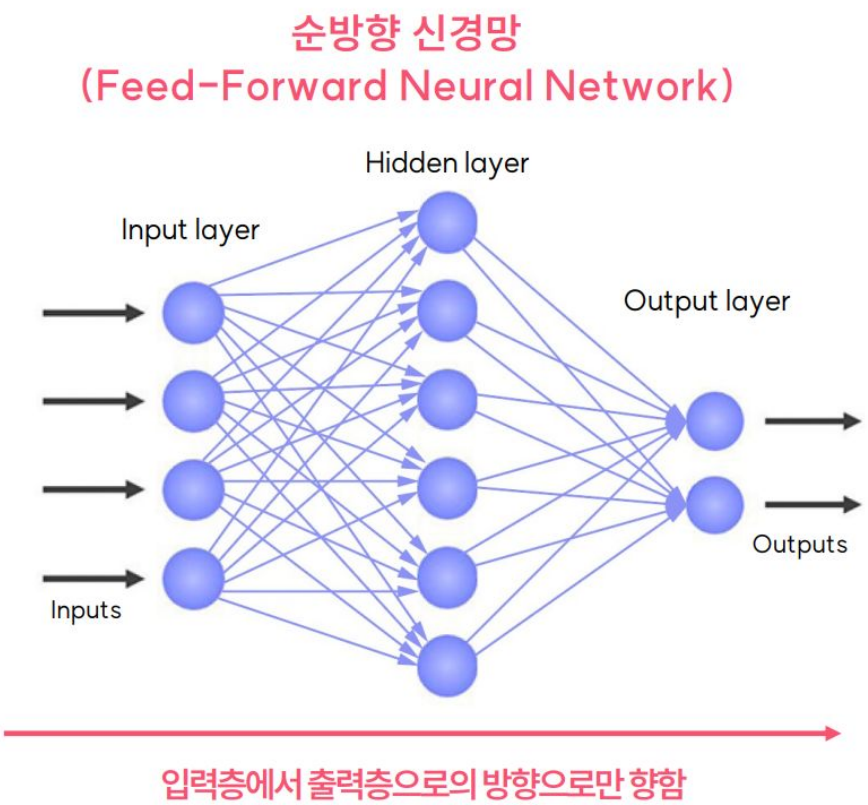
비슷하게…

Data Diving

- 단어 → 벡터 에서 벡터의 집합을 word embedding space라고 함
  - 단어를 벡터로 바꾼 걸 embedding이라고 함
- 이미지에서도 image embedding space라고 함
- space = set

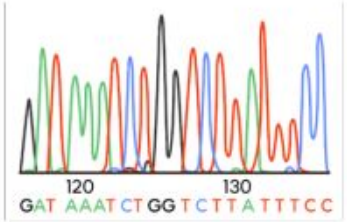
순방향 신경망의 한계

Data Diving

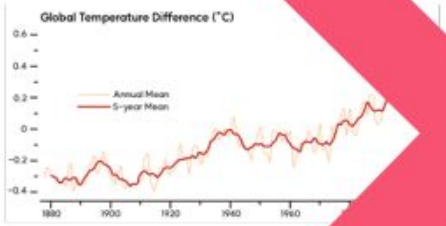


순방향 신경망의 한계

순차적 데이터를
효과적으로 이해하지 못하는 **한계 발생**



DNA 염기 서열



세계 기온

소리 신호

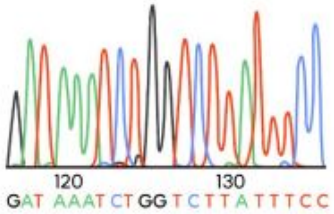
동해물과 백두산이
마르고 닳도록

텍스트

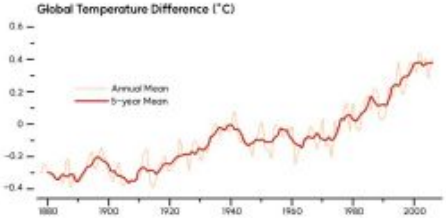
순방향 신경망의 한계

순차적 데이터
(Sequential data)

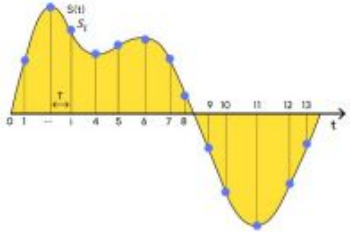
순서를 가지는 데이터의 집합



DNA 염기 서열



세계 기온 변화



샘플링된 소리 신호

동해물과 백두산이
마르고 닳도록

텍스트

순서가 바뀌면 데이터 고유의 특성을 잃어버림

순방향 신경망의 한계

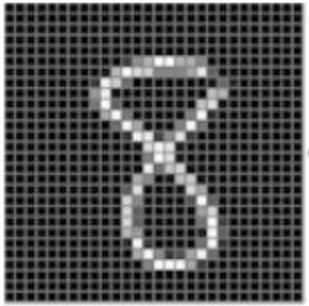
Data Diving

"그녀는 20살이고 그녀의 아버지는 50살이다"

"그녀의 아버지는 20살이고 그녀는 50살이다"

단어의 위치가 바뀌면 전체적인 의미가 왜곡됨

순방향 신경망의 한계

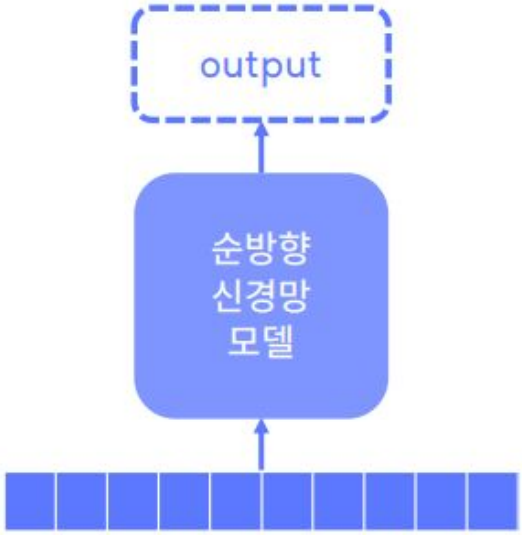


28 x 28
784 pixels

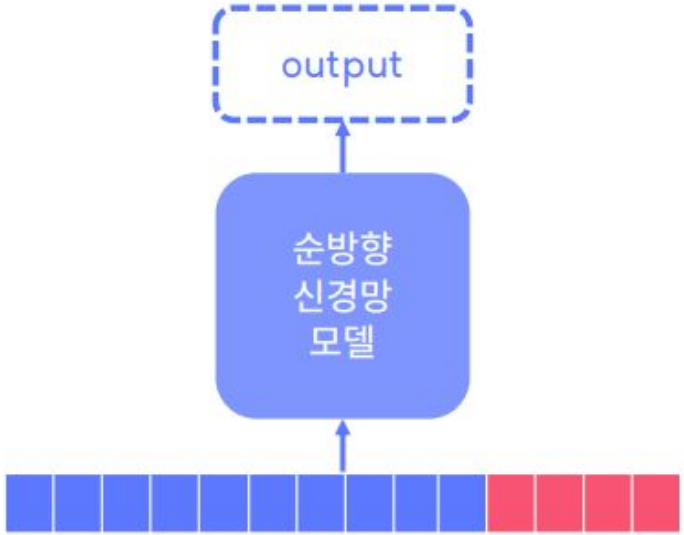
고정된 크기의 데이터
ex) MNIST 이미지(28*28)

가변적인 크기의 순차적 데이터
ex) 동영상, 에세이, 음악, 주가 차트등

순방향 신경망의 한계



크기가 딱 맞는 입력



크기를 초과한 입력
→ 작동 불가

순방향 신경망은 가변적인 크기의 데이터를 모델링하기에는 적합하지 않음

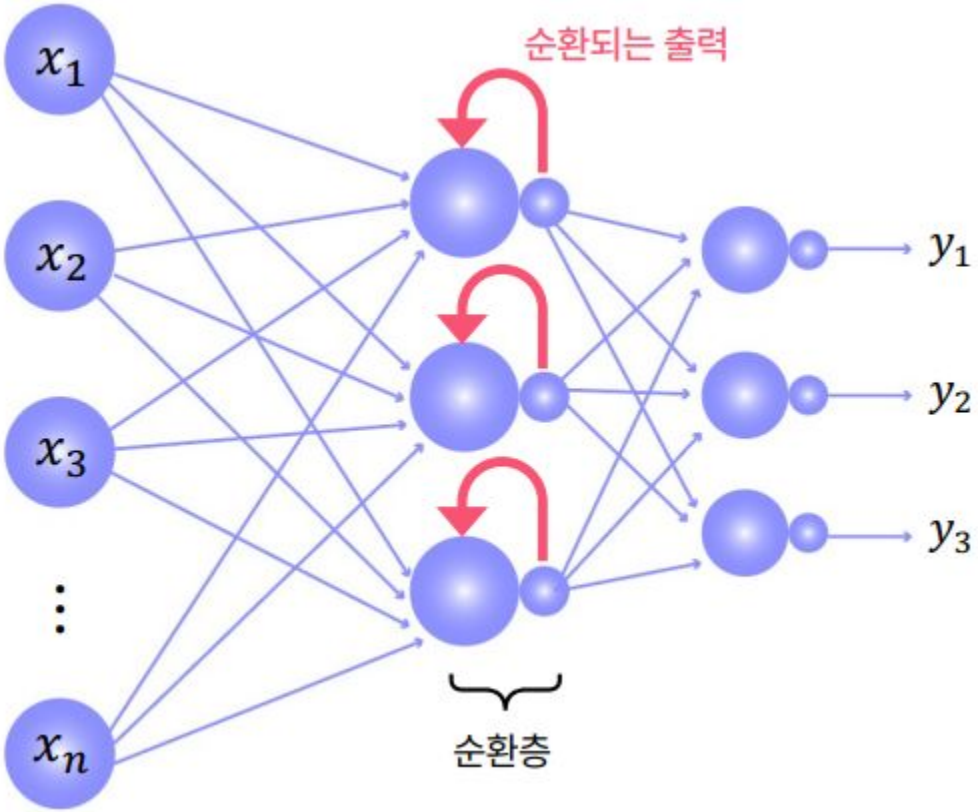
순방향 신경망의 한계

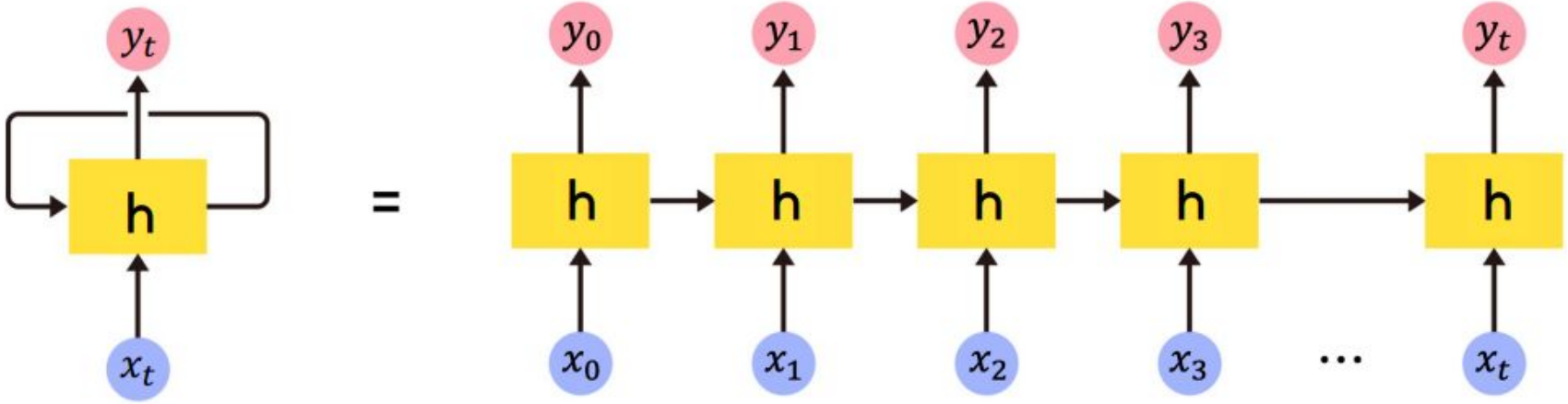
한계를 어떻게 극복할 수 있을까?

1. 데이터의 순서 정보와 이전에 입력된 데이터를 모두 기억할 수 있는 구조
2. 다양한 길이의 입력 시퀀스를 처리할 수 있는 구조

순방향 신경망의 한계


순환 신경망
RNN
Recurrent Neural Network
입력과 출력을 시퀀스(연관된 연속의 데이터) 단위로 처리하는 인공신경망


x_1
x_2
x_3
x_n
순환되는 출력
y_1
y_2
y_3
순환층



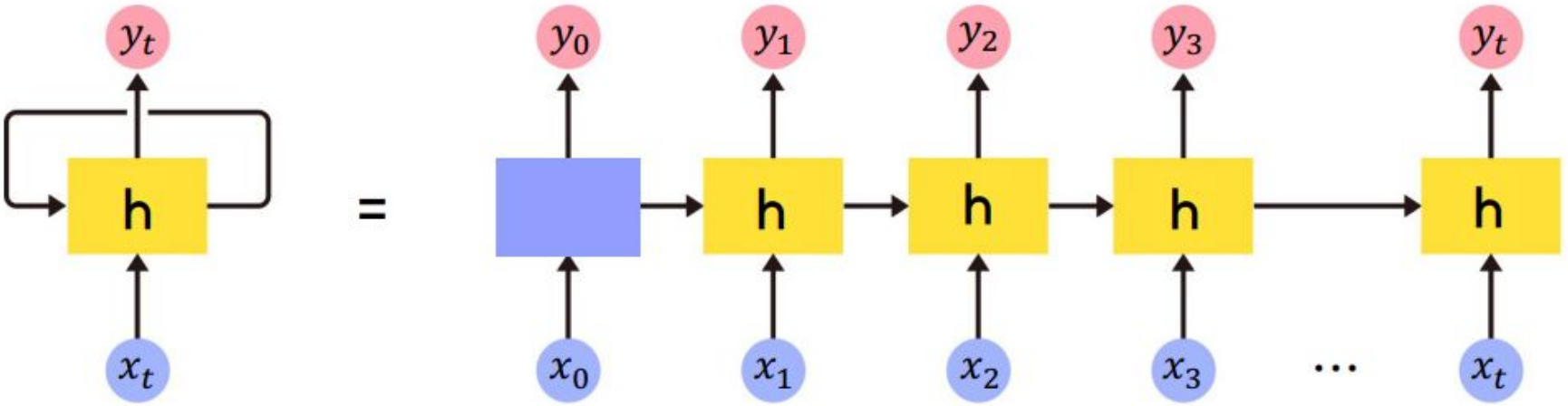
재귀 형태

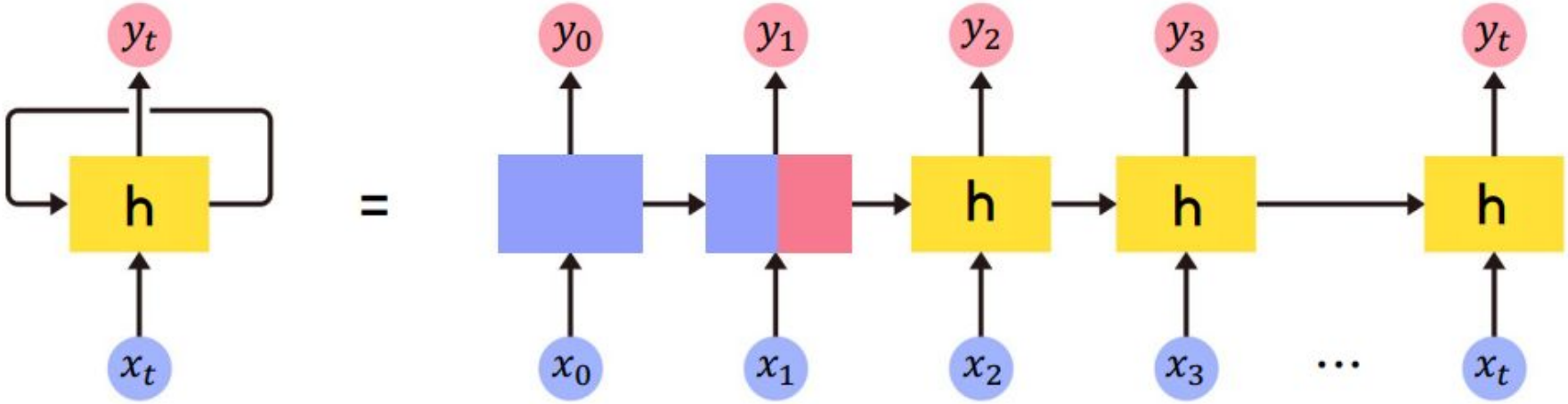
타임 스텝(time step)으로 펼쳐서 표현한 형태

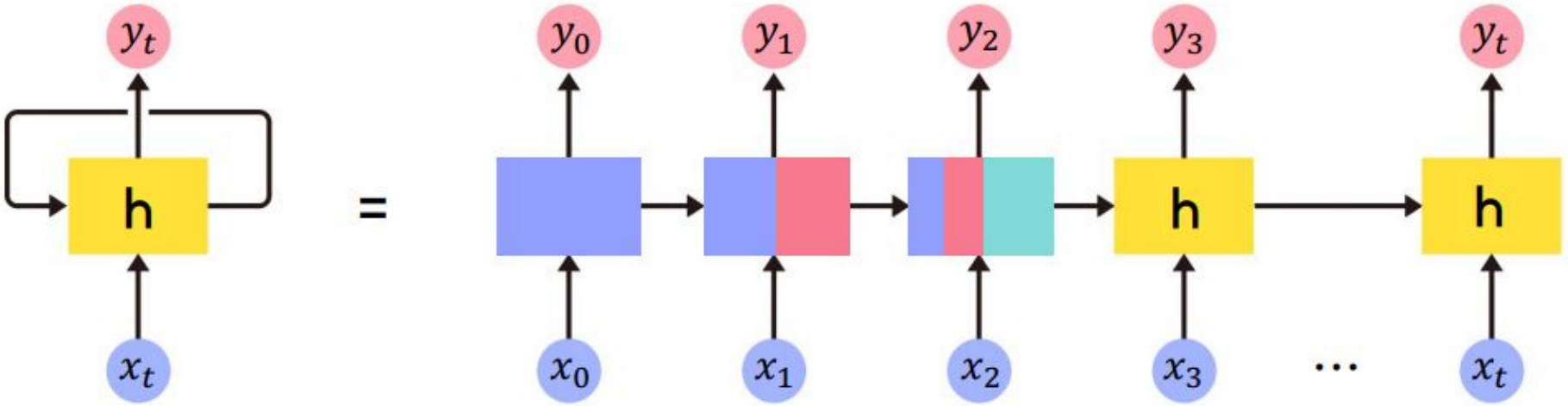
이전 단계의 결과가 다음 단계의 입력이 되는 순환적인 구조

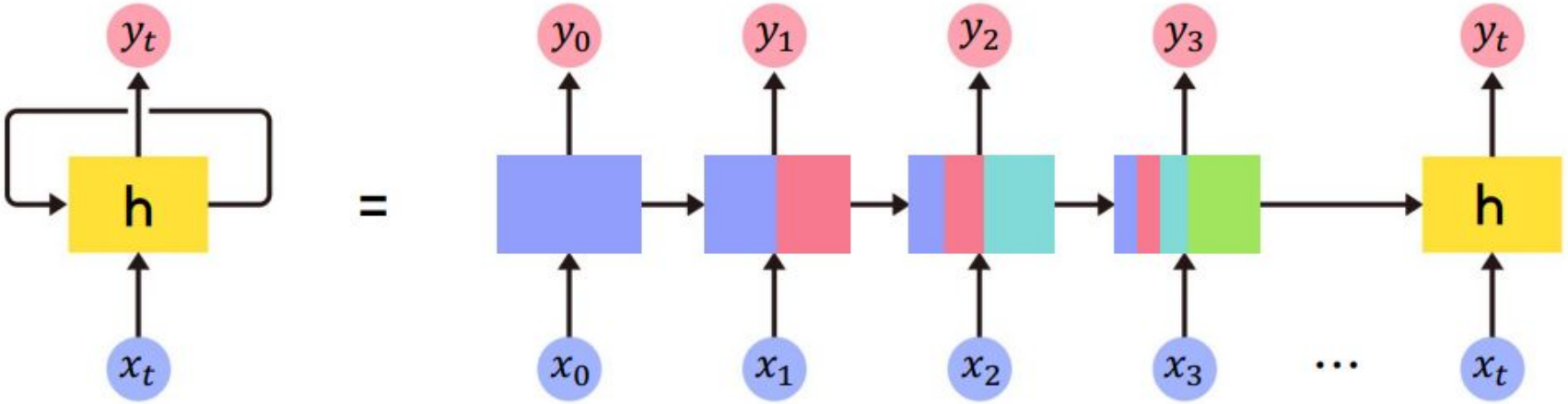
지금까지 입력된 데이터에 대한 '기억'을 가지고 있음

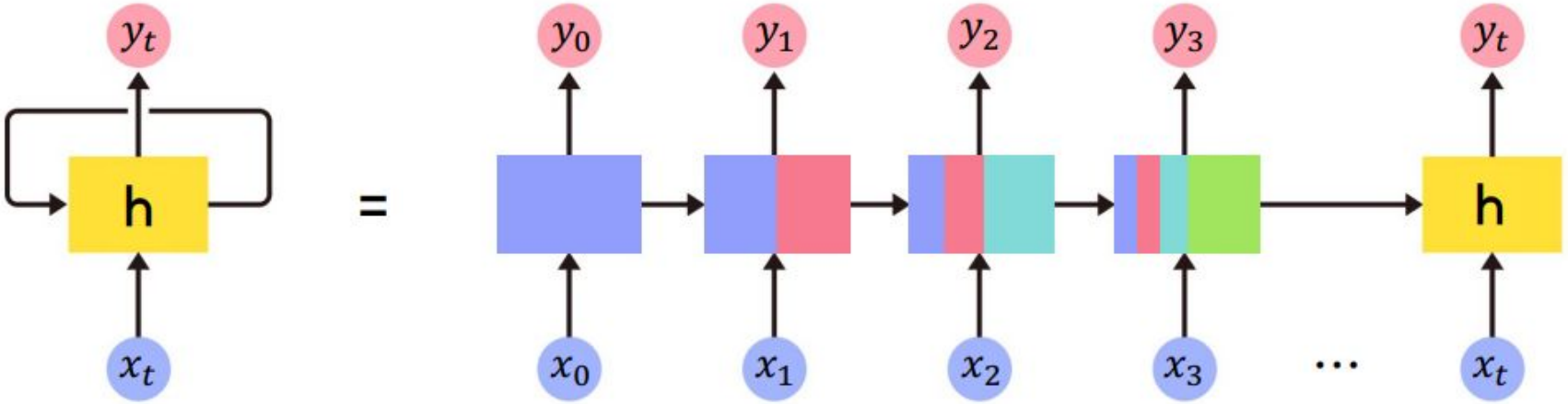
RNN

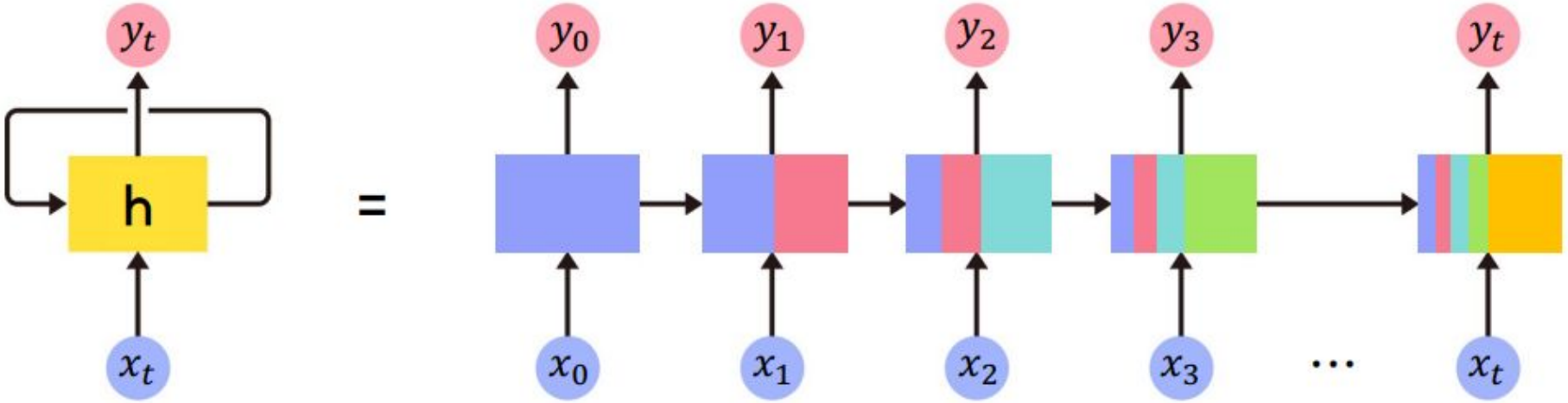



y_t
h
x_t
=
y_0
y_1
y_2
y_3
y_t
h
h
h
h
x_0
x_1
x_2
x_3
...
x_t


y_t
h
x_t
=
y_0
y_1
y_2
y_3
y_t
h
h
x_0
x_1
x_2
x_3
...
x_t


y_t
h
x_t
=
y_0
y_1
y_2
y_3
y_t
h
x_0
x_1
x_2
x_3
...
x_t


y_t
h
x_t
=
y_0
y_1
y_2
y_3
y_t
h
x_0
x_1
x_2
x_3
...
x_t


y_t
h
x_t
=
y_0
y_1
y_2
y_3
y_t
x_0
x_1
x_2
x_3
...
x_t

기억 시스템
(Memory system)

지금까지 입력된 데이터를 기억하여

새로운 입력이 들어올 때 자신의 기억을 조금씩 수정하며 누적하는 시스템

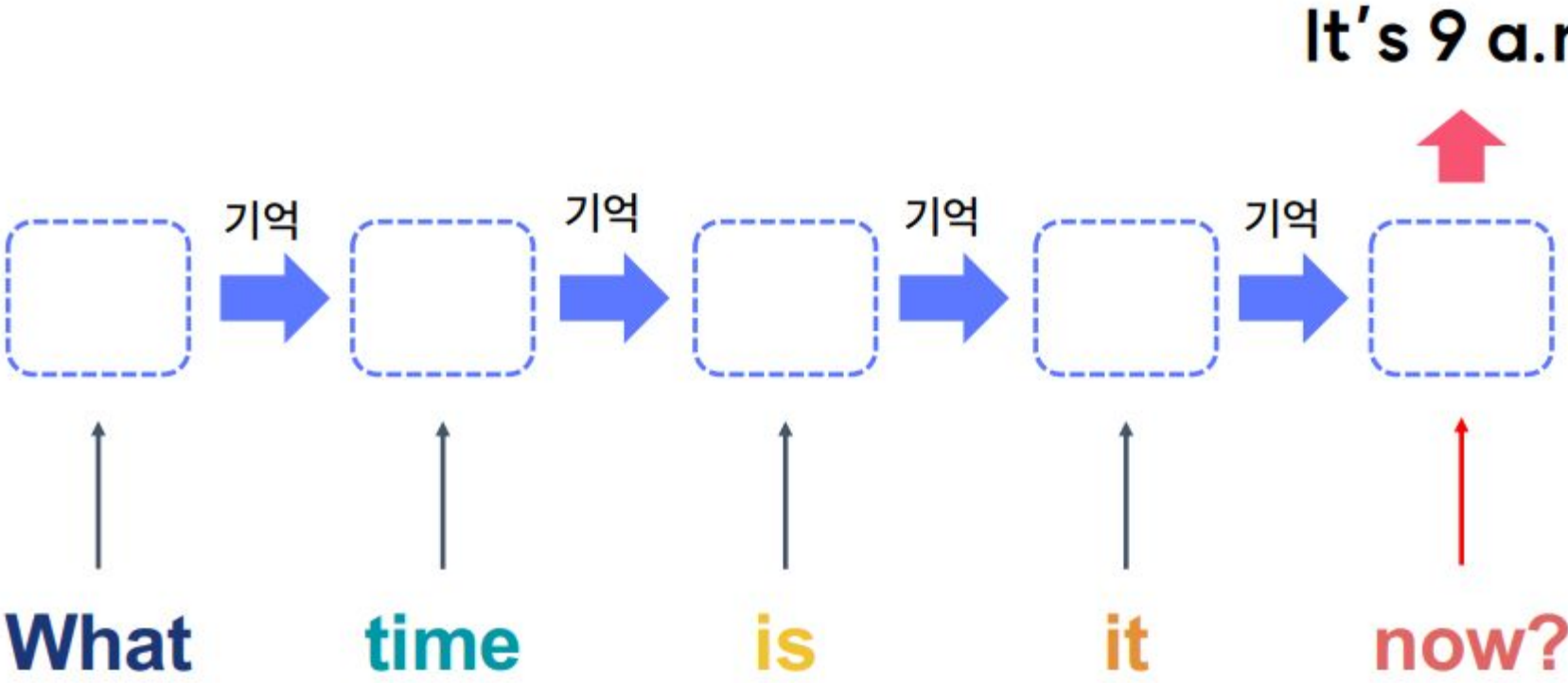
네트워크 안의 루프(loop)를 통해 정보가 지속되도록 함

→ 과거의 기억을 토대로 미래를 예측할 수 있음

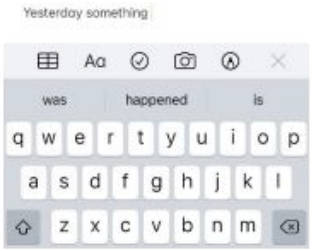
RNN이 순환적(recurrent)하다고 불리는 이유?

1. 시퀀스의 모든 요소에 대해 동일한 작업을 반복 수행

2. 출력값이 이전의 계산 결과에 영향을 받음


It's 9 a.m
기억
기억
기억
기억
What
time
is
it
now?

① Language modeling



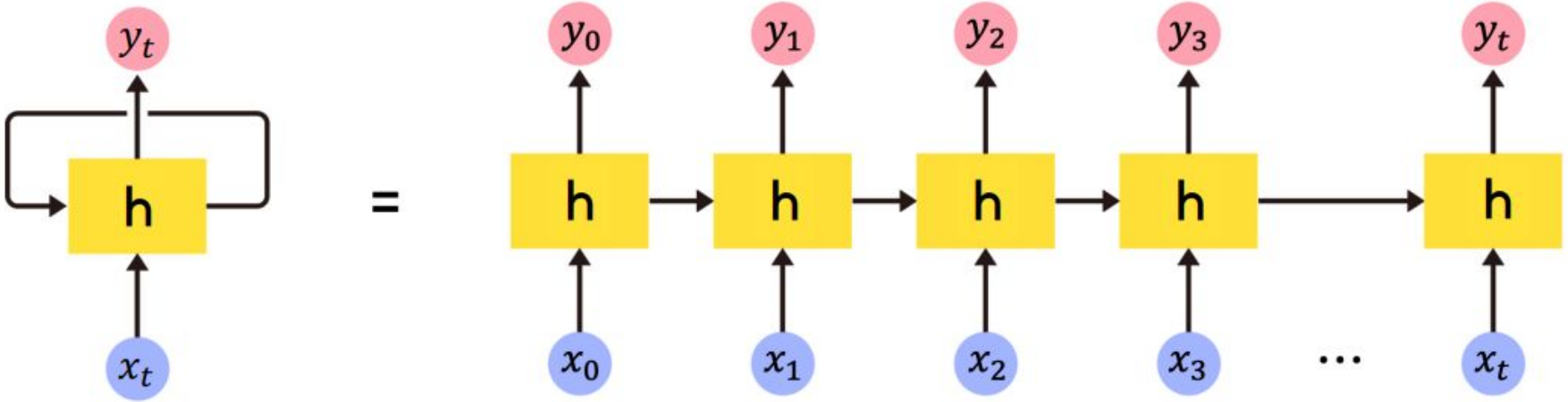
② Speech recognition

③ Machine translation

papago

④ Image captioning

→ ①언어 모델링(Language modeling), ②음성 인식(Speech recognition),
③기계 번역(Machine translation), ④이미지 캡셔닝(Image captioning) 등의 많은 응용분야에 사용



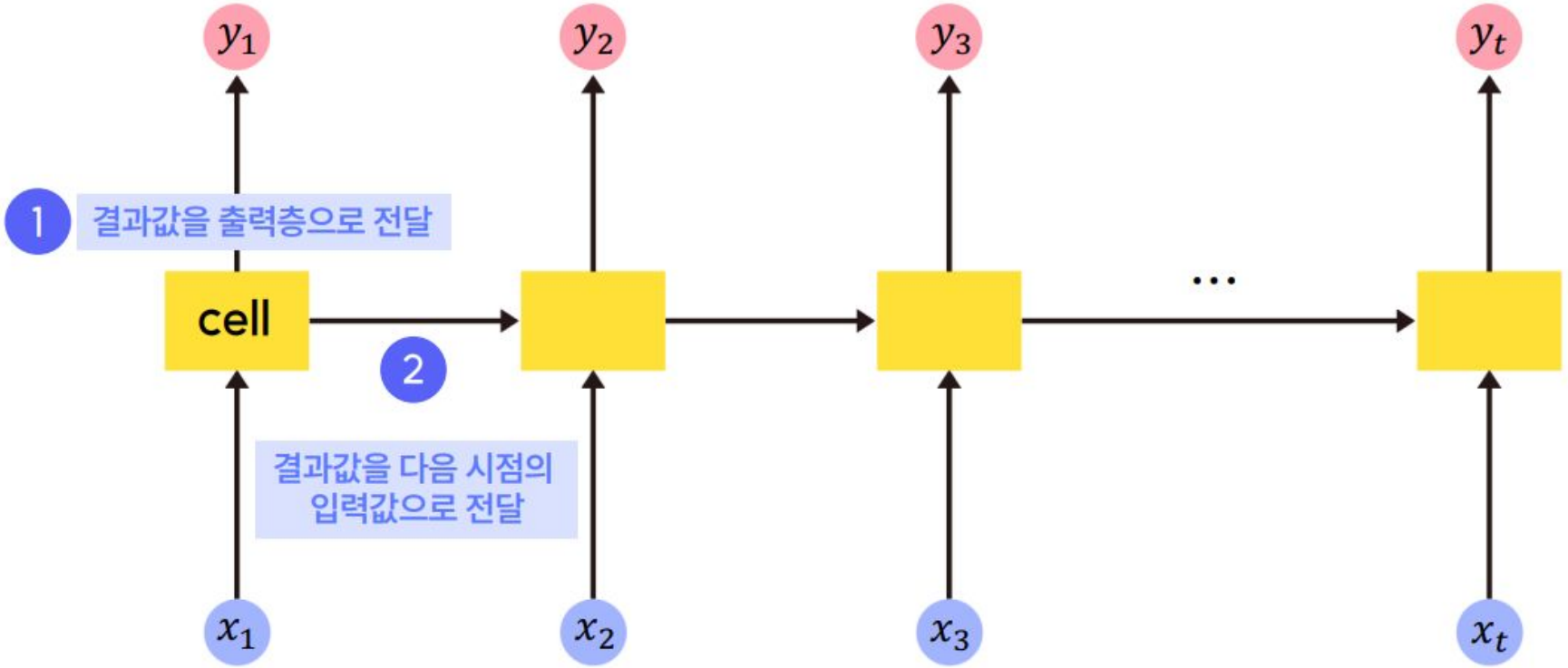
재귀 형태

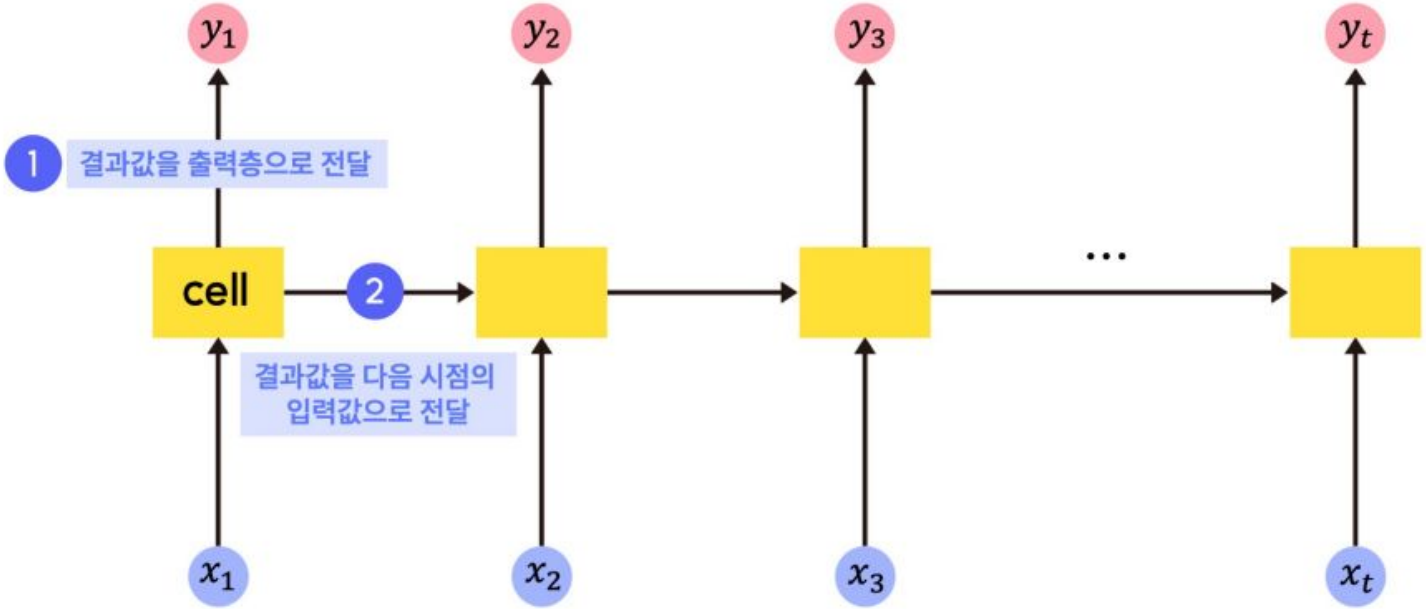
타임 스텝(time step)으로 펼쳐서 표현한 형태

이전 단계의 결과가 다음 단계의 입력이 되는 순환적인 구조

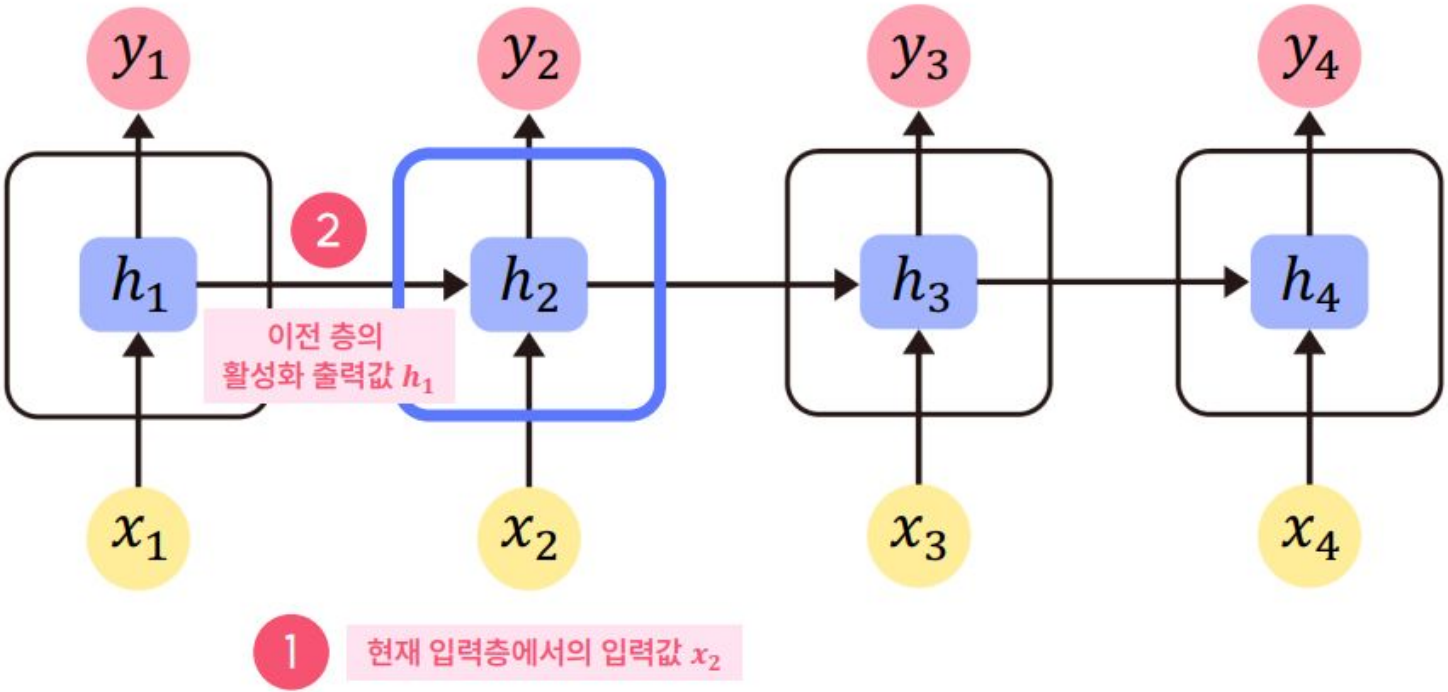
지금까지 입력된 데이터에 대한 '기억'을 가지고 있음


시점(time step)
...
이전 시점(t-1)
현재 시점(t)
다음 시점(t+1)
...

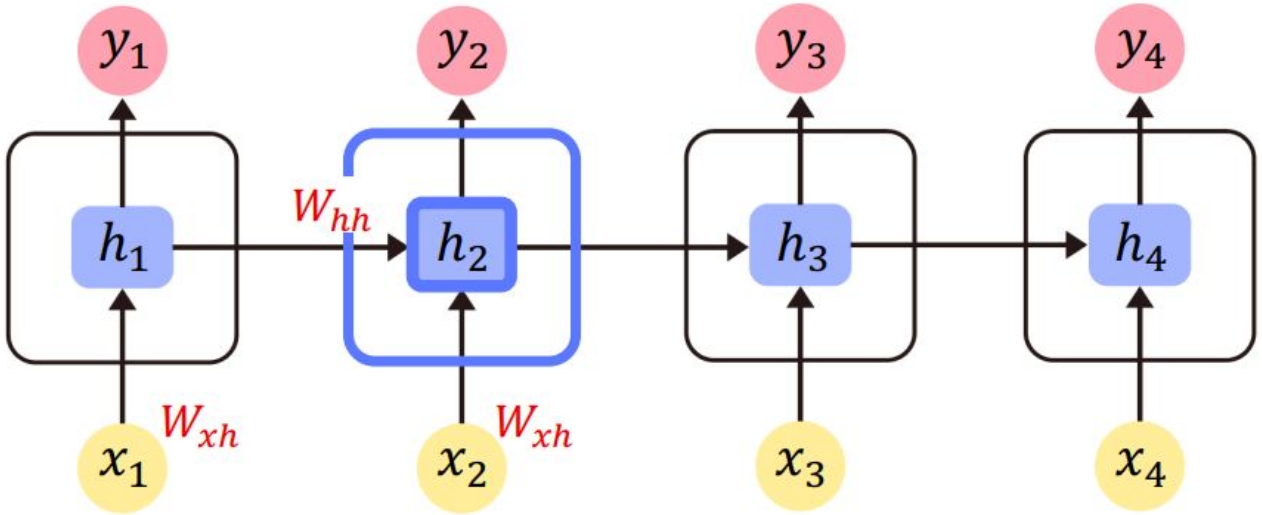

y_1
y_2
y_3
y_t
1
결과값을 출력층으로 전달
cell
2
결과값을 다음 시점의
입력값으로 전달
...
x_1
x_2
x_3
x_t



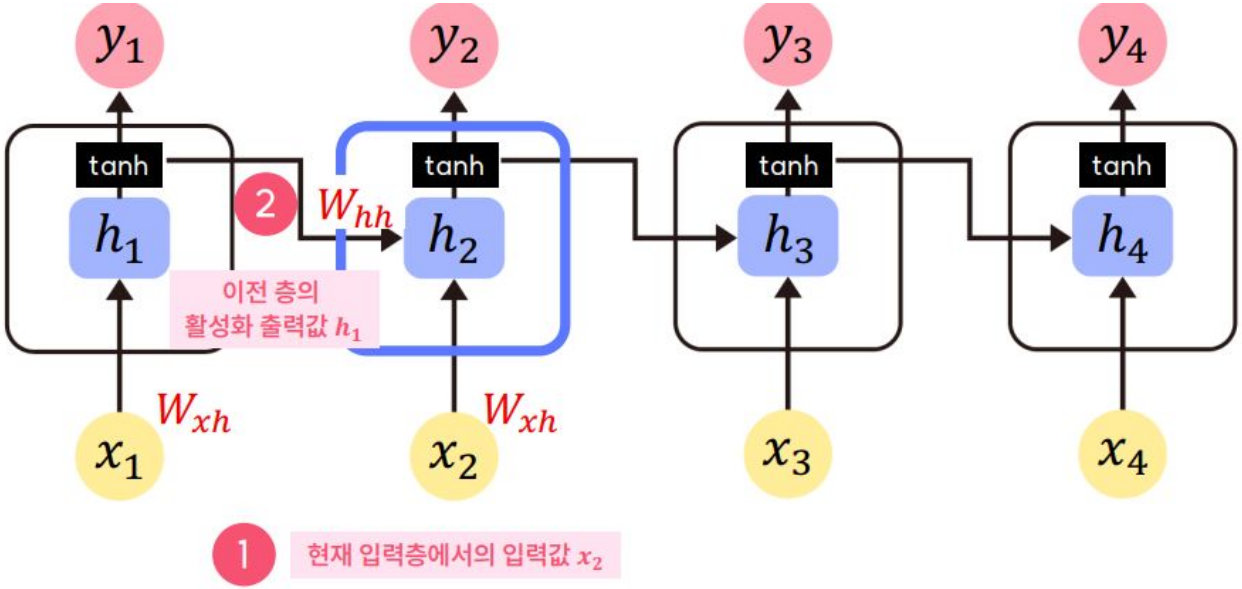
hidden state(은닉 상태) : cell의 출력값, cell이 다음 시점(t+1)에서 자신에게 보내는 값



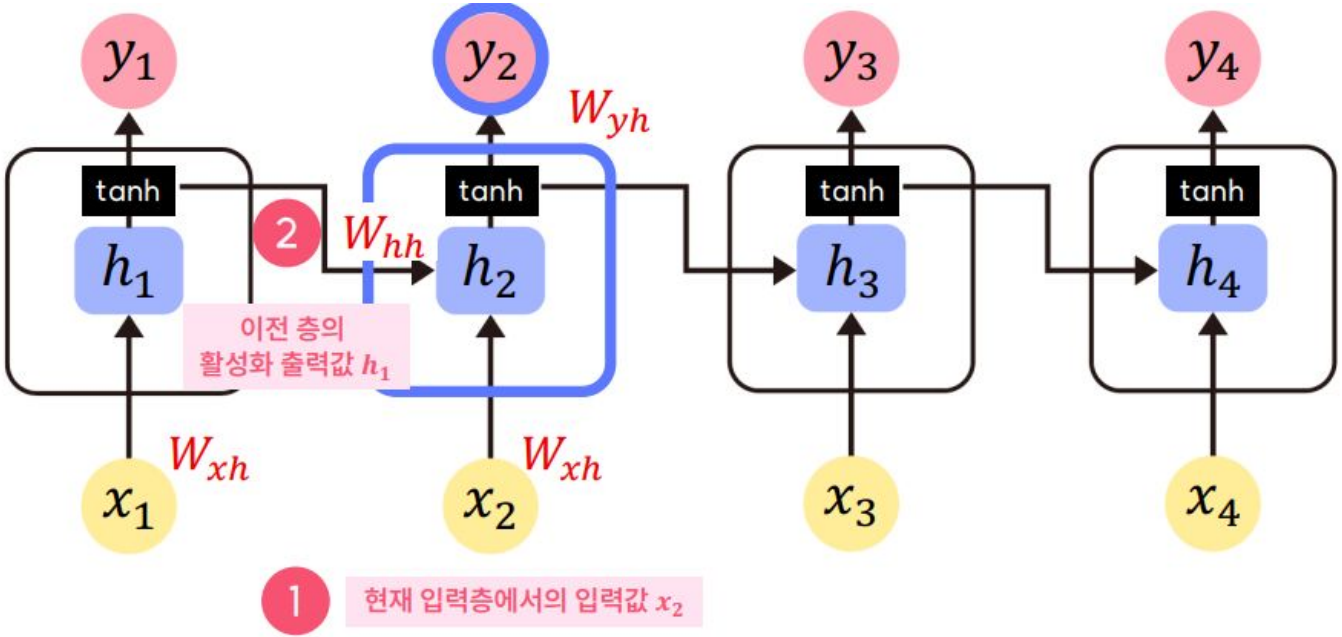
RNN은 2개의 입력(현재 입력층에서의 입력값, 이전 층의 활성화 출력값)이 존재



(현재 입력층에서의 입력값 x_2)*(입력층에서의 가중치 W_{xh})+(이전 층의 활성화 출력값 h_1*전달 시의 가중치 W_{hh})

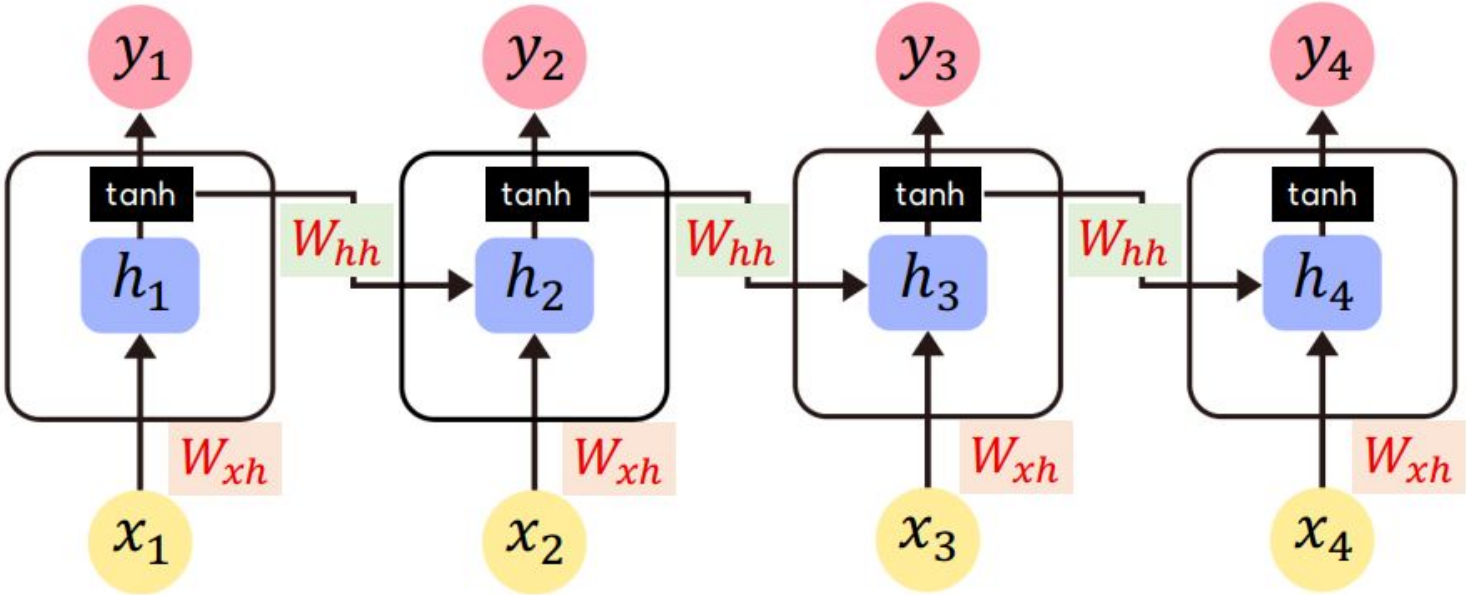


h_2 = Tanh{(현재 입력층에서의 입력값 x_2)+(*입력층에서의 가중치 W_{xh})+(이전 층의 활성화 출력값 h_1*전달 시의 가중치 W_{hh})}



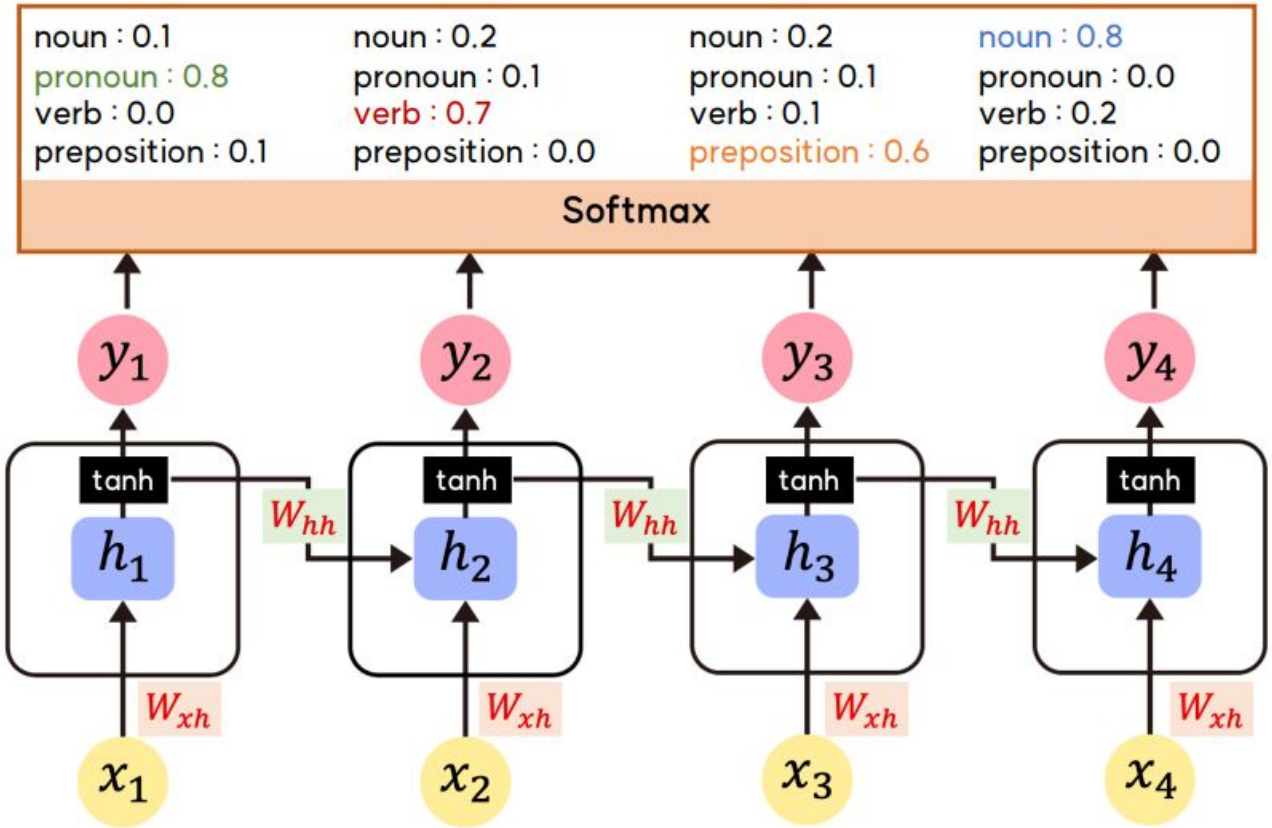
y_2 = W_{yh}[Tanh{(현재 입력층에서의 입력값 x_2)+(*입력층에서의 가중치 W_{xh})+(이전 층의 활성화 출력값 h_1 *전달 시의 가중치 W_{hh})}]

$= h_2$



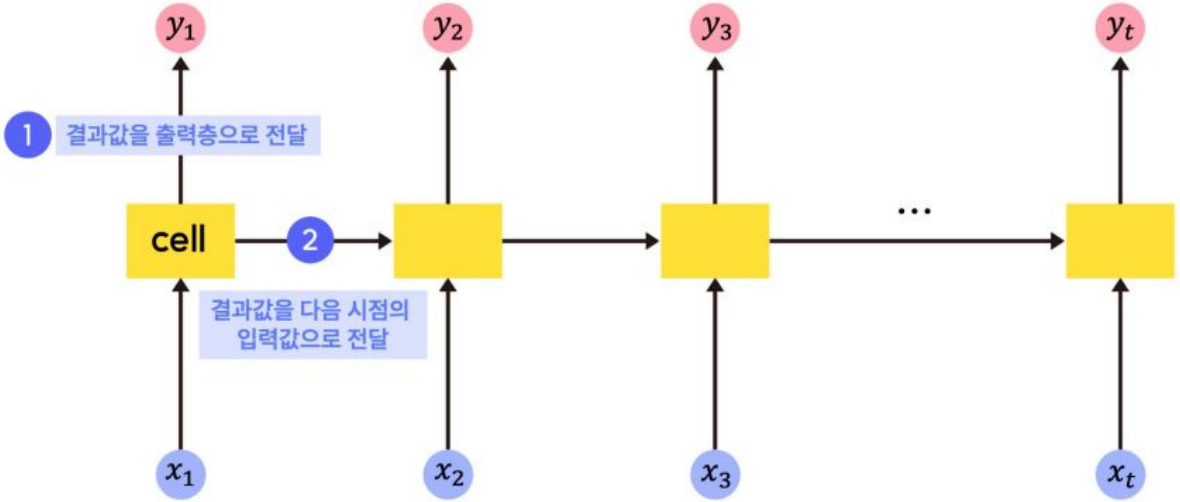
W_{hh}, W_{xh}와 활성화 함수는 모든 셀에서 동일한 값을 가짐

→ 학습해야 하는 파라미터 수 대폭 감소


noun : 0.1
pronoun : 0.8
verb : 0.0
preposition : 0.1
noun : 0.2
pronoun : 0.1
verb : 0.7
preposition : 0.0
noun : 0.2
pronoun : 0.1
verb : 0.1
preposition : 0.6
noun : 0.8
pronoun : 0.0
verb : 0.2
preposition : 0.0
Softmax
y_1
y_2
y_3
y_4
tanh
tanh
tanh
tanh
h_1
h_2
h_3
h_4
W_{hh}
W_{hh}
W_{hh}
W_{xh}
W_{xh}
W_{xh}
W_{xh}
x_1
x_2
x_3
x_4

RNN

Data Diving



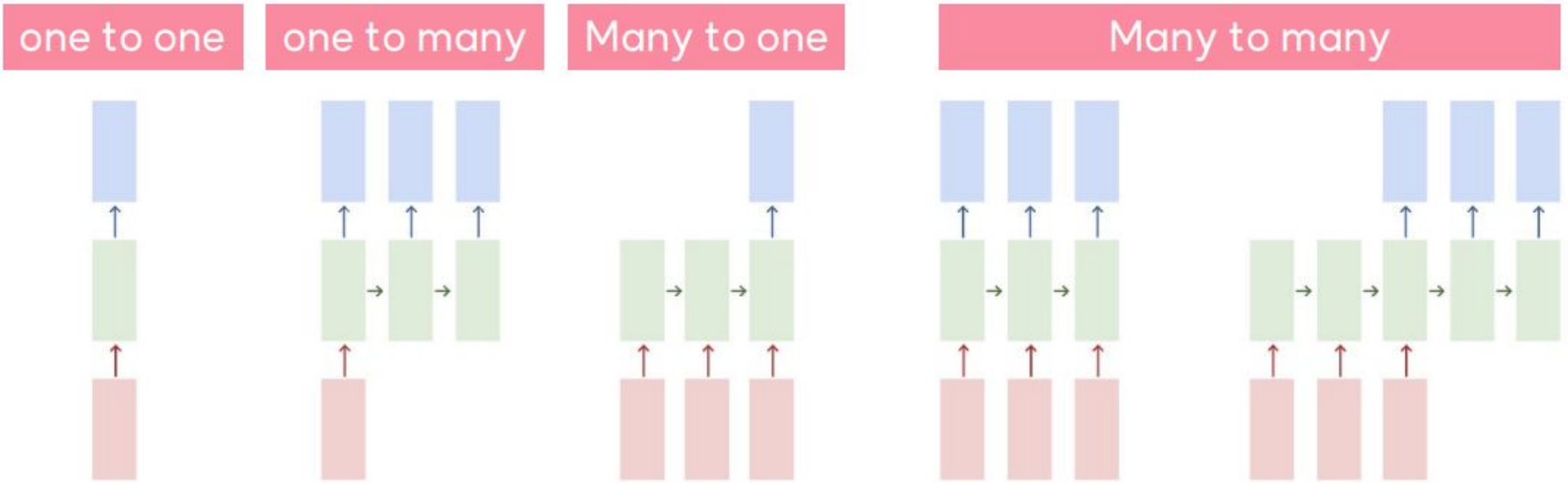
1. RNN 셀은 동일한 셀로, 전체 모델에서 재사용됨

2. RNN은 모든 단계에서 파라미터 값(W_{hh} , W_{xh})과 활성화 함수를 공유함

3. RNN의 출력은 이전의 모든 입력의 영향을 받음

RNN

Data Diving



RNN

input

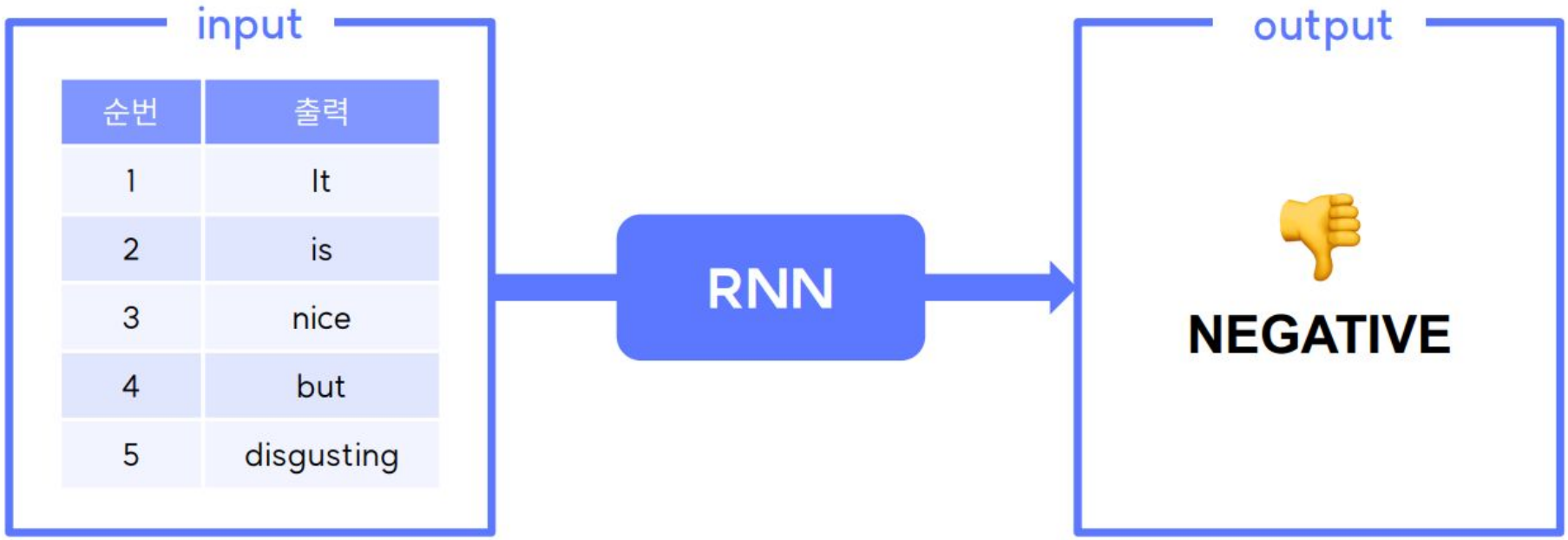
RNN

output

순번	출력
1	He
2	is
3	playing
4	with
5	basketball

하나의 이미지 입력에 대해 사진의 제목을 출력하는 이미지 캡셔닝
(image captioning)

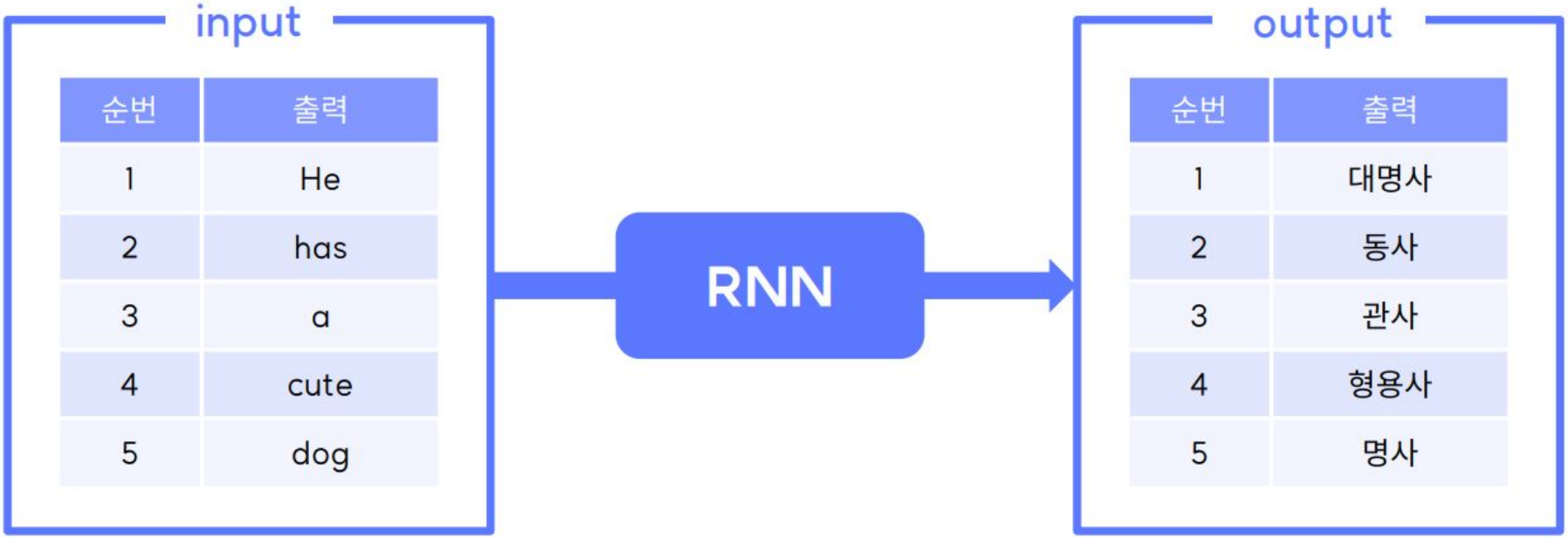
사진 → 단어들



입력이 긍정인지 부정인지 판단하는 감정 분류
(sentiment classification)

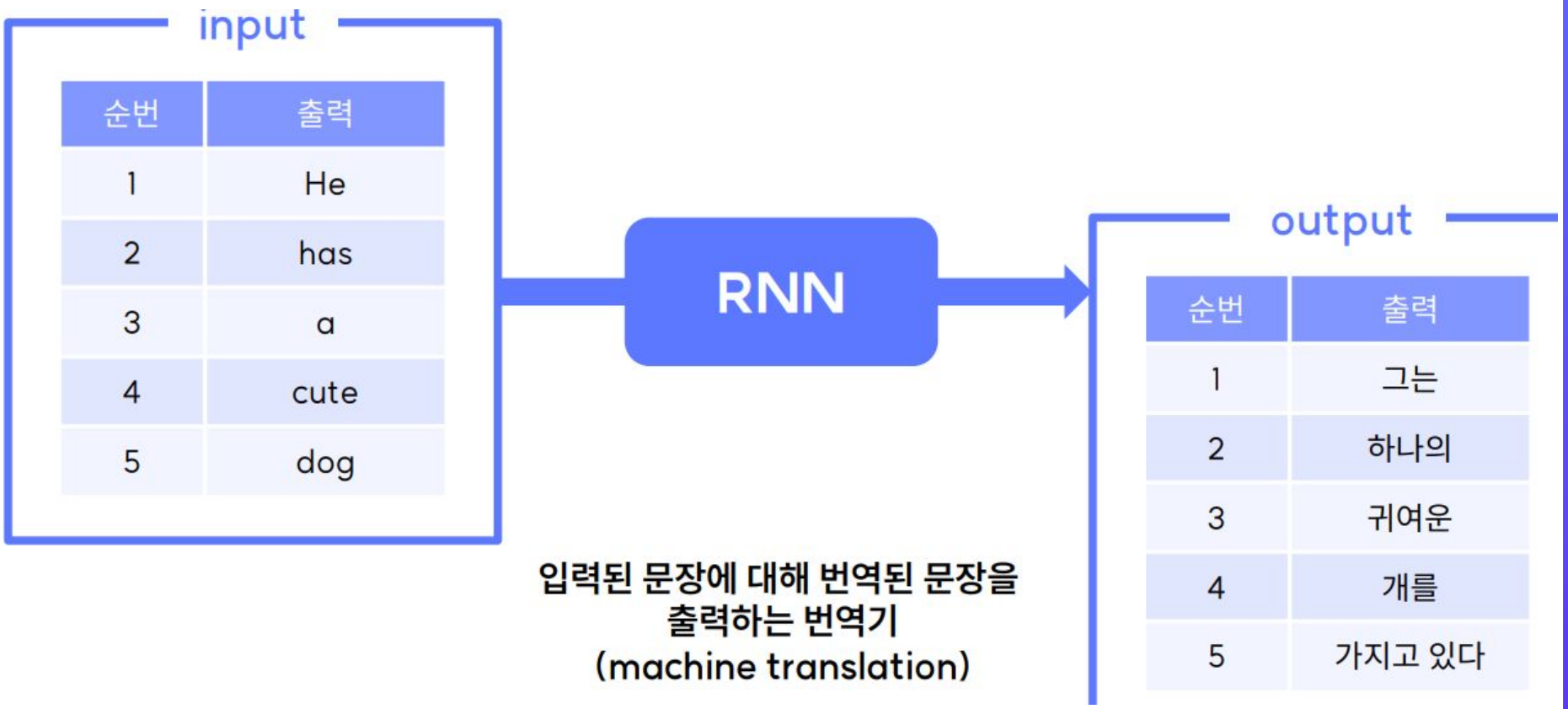
단어들 → 감정

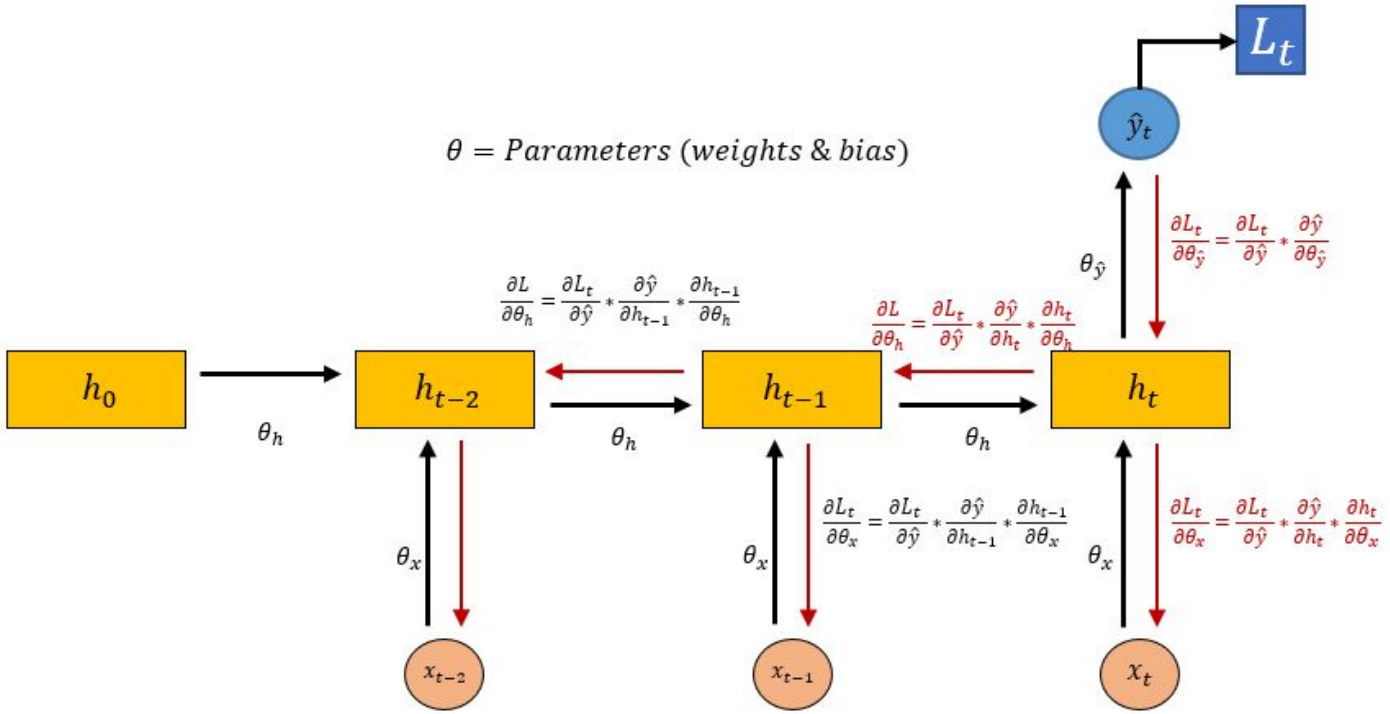
RNN



입력된 문장에 포함된 단어의 품사를 예측하는 품사 태깅
(POS tagging)

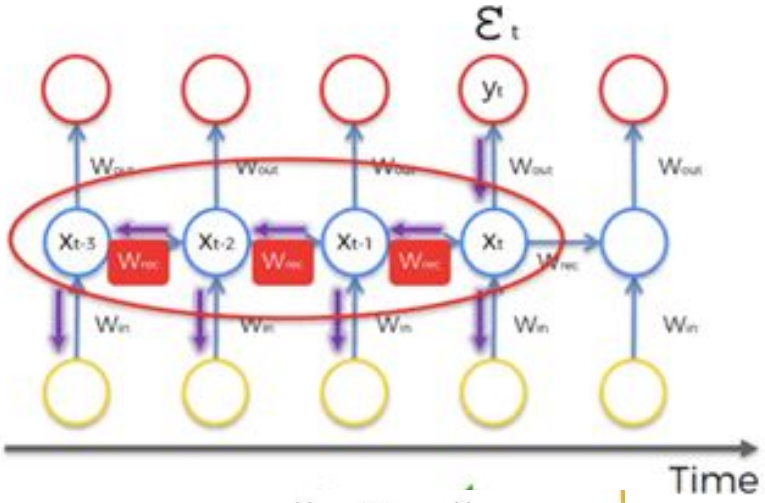
단어들 → 단어들


input
순번
출력
1
He
2
has
3
a
4
cute
5
dog
RNN
output
순번
출력
1
그는
2
하나의
3
귀여운
4
개를
5
가지고 있다
입력된 문장에 대해 번역된 문장을
출력하는 번역기
(machine translation)



Gradient Backpropagation Through Time

The Vanishing Gradient Problem



$$\frac{\partial \mathcal{E}}{\partial \theta} = \sum_{1 \le t \le T} \frac{\partial \mathcal{E}_t}{\partial \theta} \quad (3)$$

$$\frac{\partial \mathcal{E}_t}{\partial \theta} = \sum_{1 \le k \le t} \left(\frac{\partial \mathcal{E}_t}{\partial \mathbf{x}_t} \frac{\partial \mathbf{x}_t}{\partial \mathbf{x}_k} \frac{\partial^+ \mathbf{x}_k}{\partial \theta} \right) \quad (4)$$

$$\frac{\partial \mathbf{x}_t}{\partial \mathbf{x}_k} = \prod_{t \ge i > k} \frac{\partial \mathbf{x}_i}{\partial \mathbf{x}_{i-1}} = \prod_{t \ge i > k} \mathbf{W}_{rec}^T diag(\sigma'(\mathbf{x}_{i-1})) \quad (5)$$

W_{rec} ~ small ⇨ Vanishing

W_{rec} ~ large ⇨ Exploding

1. *Vanishing gradient* $\left\| \frac{\partial h_i}{\partial h_{i-1}} \right\|_2 < 1$

2. *Exploding gradient* $\left\| \frac{\partial h_i}{\partial h_{i-1}} \right\|_2 > 1$

Formula Source: Razvan Pascanu et al. (2013)