KUBIG 딥러닝 분반 (6주차)

Recurrent Neural Networks

딥러닝 분방장: 김태영, 오화진

01 sequencial data

02 statistical tool

03 text data preprocess -ing

04 language model

05 RNN

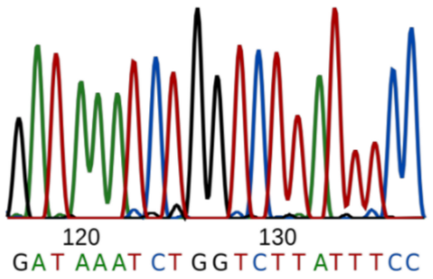
06 week 6 과제

딥러닝 분방장 : 김태영, 오화진

01. sequencial data

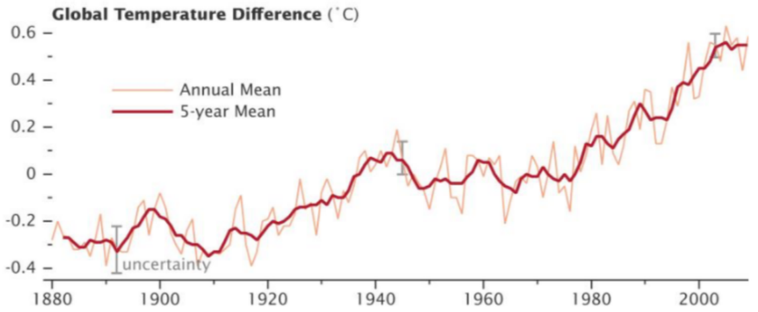
01. sequencial data

순차 데이터 Sequential Data



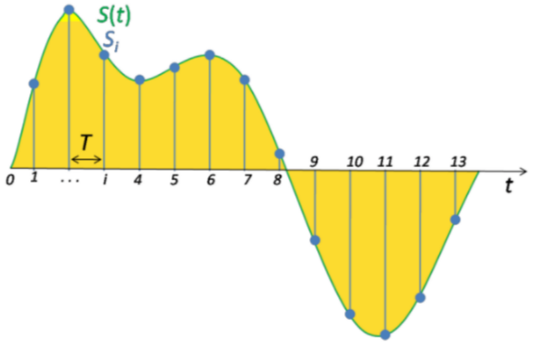
DNA 염기 서열

(Sequential Data)



세계 기온 변화

(Temporal Sequence)



샘플링된 소리 신호

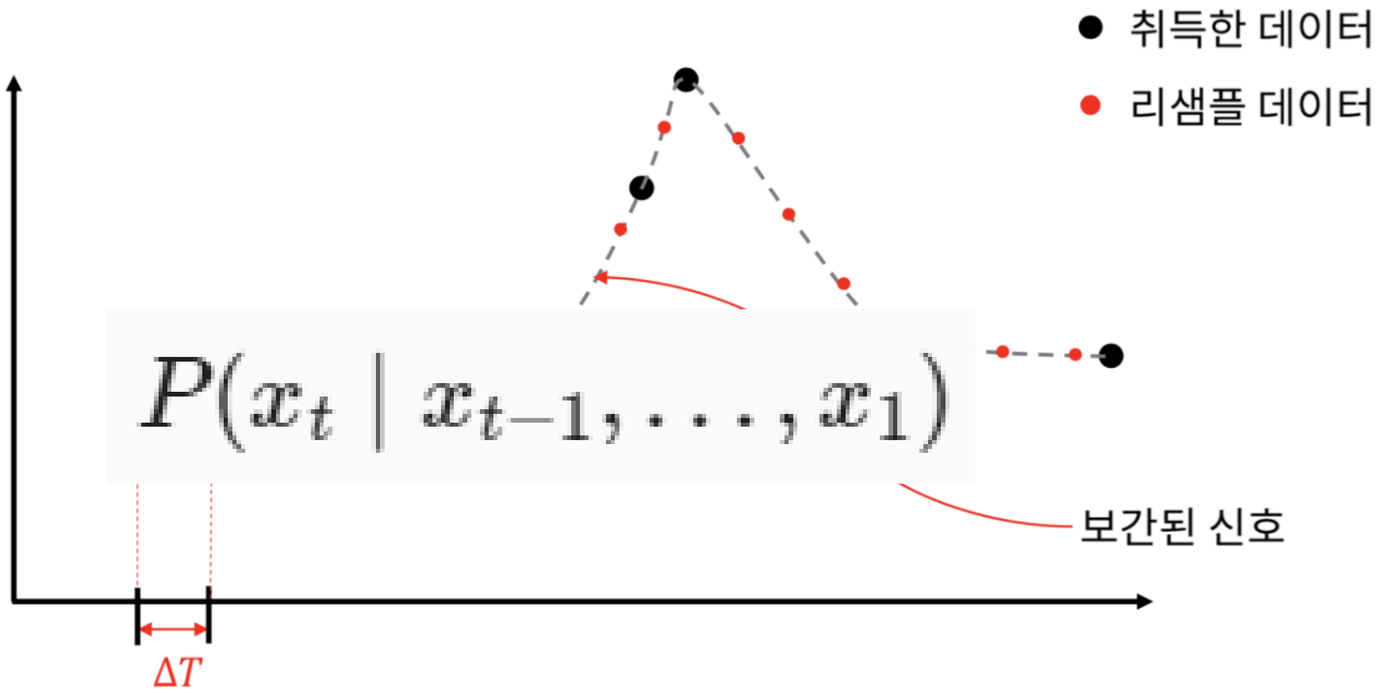
(Time Series)

순서가 의미가 있으며, 순서가 달라질 경우 의미가 손상되는 데이터를 순차 데이터라고 한다.

시간적 의미가 있는 경우 Temporal Sequence라고 하며, 일정한 시간차라면 Time Series라고 한다.

01. sequencial data

Resampling



Temporal Sequence를 Time Series로 변환하기 위해서는 Resample을 수행한다.

취득된 데이터(Temporal Sequence)를 이용해 신호를 보간(Interpolation)하고,

이를 균일 시간 간격으로 샘플링한다.

01. sequencial data

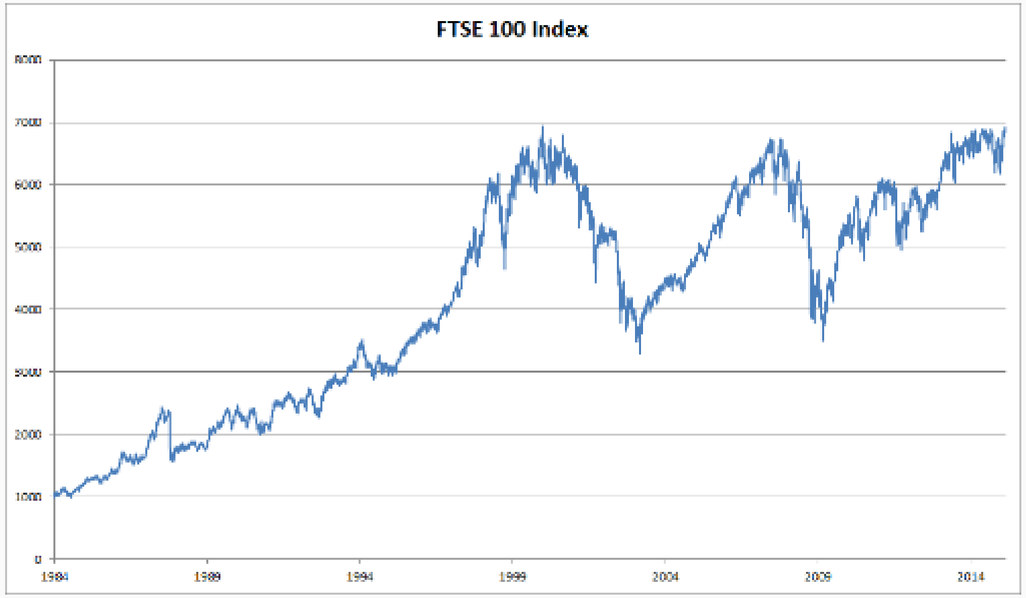


Fig. 9.1.1 FTSE 100 index over about 30 years.

$$P(x_t \mid x_{t-1}, \ldots, x_1)$$

02. statistical tool

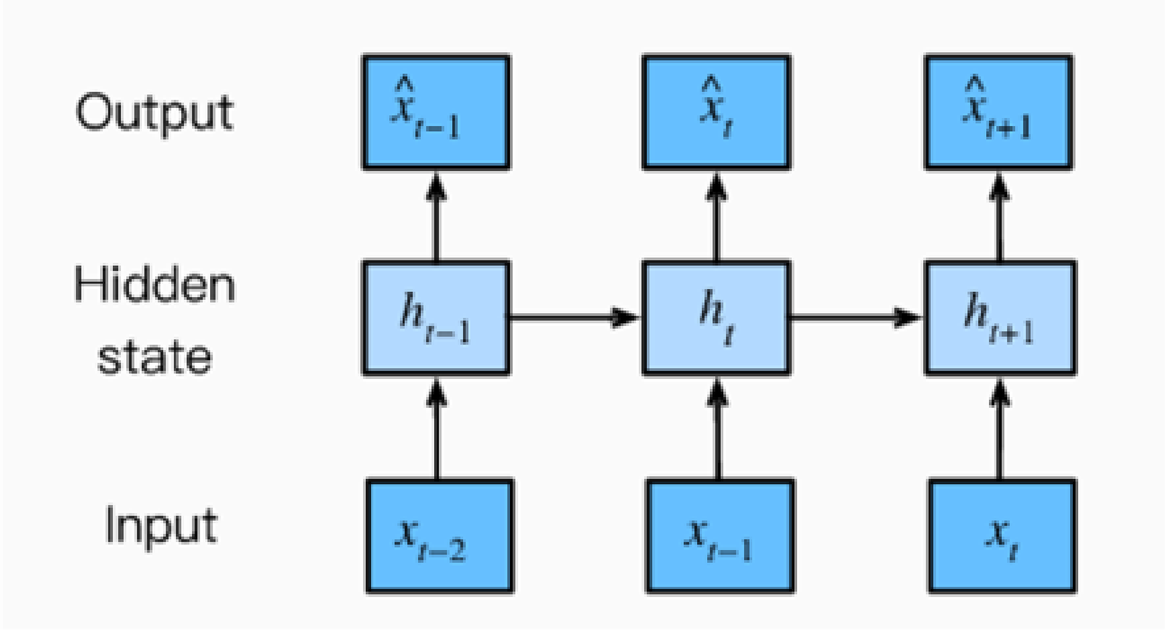
Autoregressive model

- 특정 시점의 값을 예측하는 것 $P(x_t \mid x_{t-1}, \ldots, x_1)$

- 효과적인 예측을 위해 timespan of length를 지정 (γ)

- Summary of past observation: hidden state

Latent autoregressive model



Markov models

$$O = (o_1, o_2, \cdots, o_{t-1}, o_t, o_{t+1}, \cdots, o_T)^T = o_1 o_2 \cdots o_{t-1} o_t o_{t+1} \cdots o_T$$

$$r = 0: \ P(o_t | o_{t-1} o_{t-2} \cdots o_1) = P(o_t)$$

$$r = 1: \ P(o_t | o_{t-1} o_{t-2} \cdots o_1) = P(o_t | o_{t-1})$$

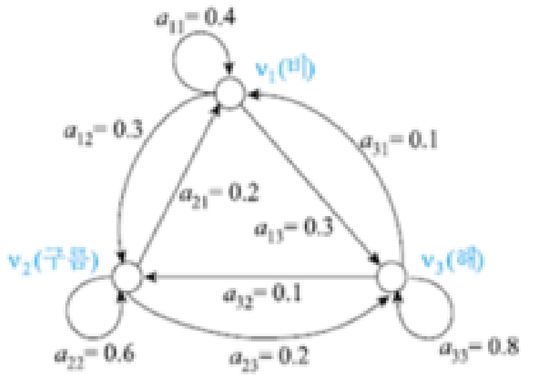
$$r = 2: \ P(o_t | o_{t-1} o_{t-2} \cdots o_1) = P(o_t | o_{t-1} o_{t-2})$$

Markov models

오늘 \ 내일	비	구름	해
비	0.4	0.3	0.3
구름	0.2	0.6	0.2
해	0.1	0.1	0.8

$$A = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = \begin{vmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{vmatrix}$$

(a) 상태 전이 확률 행렬



(b) 상태 전이도

Markov models

오늘 \ 내일	비	구름	해
비	0.4	0.3	0.3
구름	0.2	0.6	0.2
해	0.1	0.1	0.8

$$A = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = \begin{vmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{vmatrix}$$

(a) 상태 전이 확률 행렬

$$P(\text{해해해비비해구름해}) = P(\text{해})P(\text{해}|\text{해})P(\text{해}|\text{해})P(\text{비}|\text{해})P(\text{비}|\text{비})P(\text{해}|\text{비})P(\text{구름}|\text{해})P(\text{해}|\text{구름})$$

$$= 1 * a_{33} * a_{33} * a_{31} * a_{11} * a_{13} * a_{32} * a_{23}$$

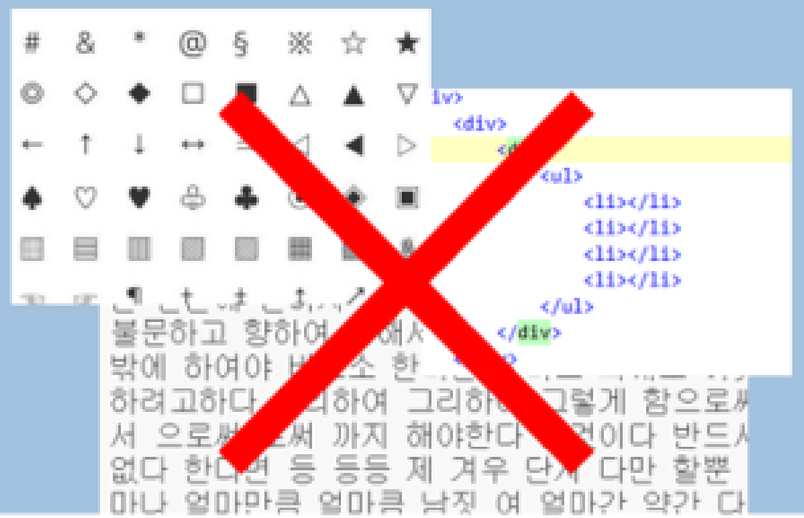
$$= 1 * 0.8 * 0.8 * 0.1 * 0.4 * 0.3 * 0.1 * 0.2$$

$$= 1.536 * 10^{-4}$$

03. text data preprocessing

클렌징 (cleansing)

**텍스트를 변환할 때 방해되거나 의미가 없어서
불필요한 것들을 제거하는 작업**



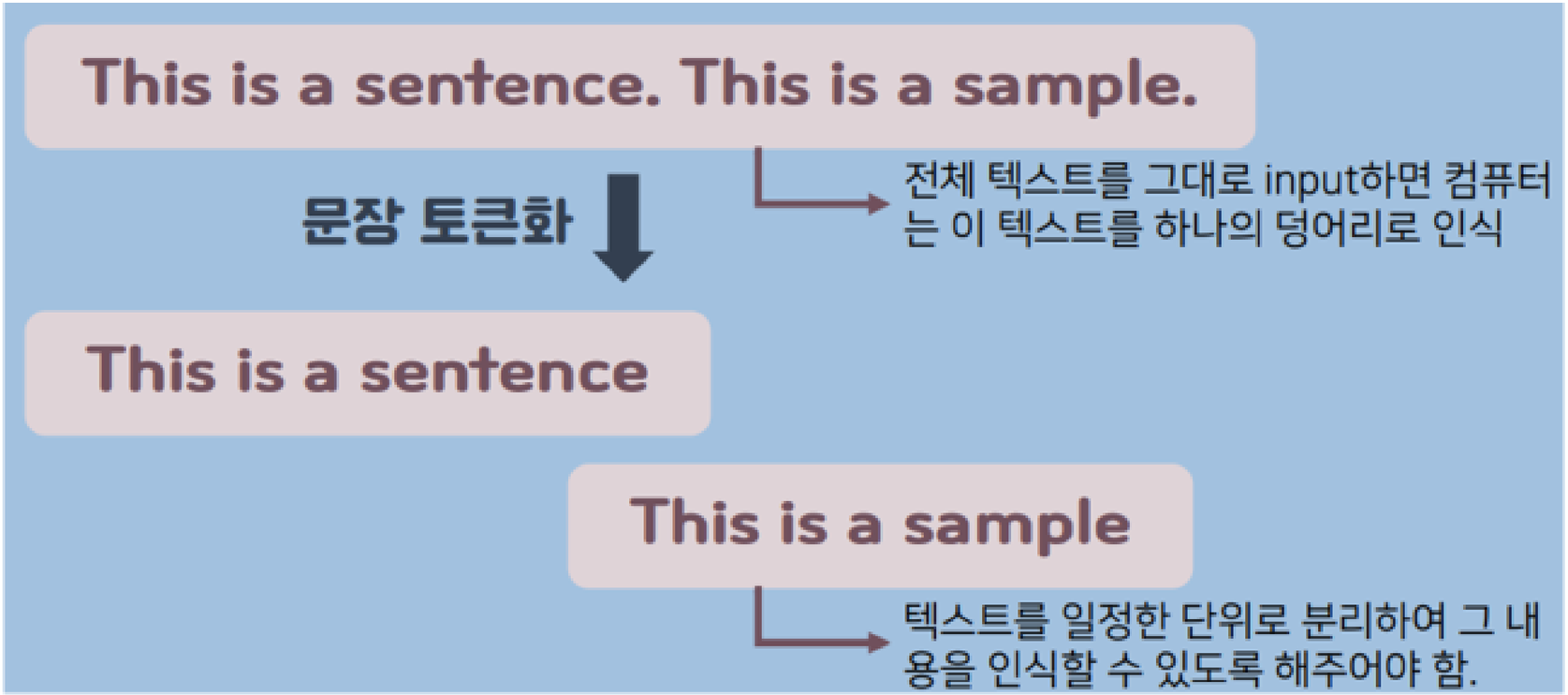
클렌징의 예시

- 특수문자, 기호 제거
- HTML 태그 제거(HTML 문서의 경우)
- 대소문자 구분 제거
- stopword 제거

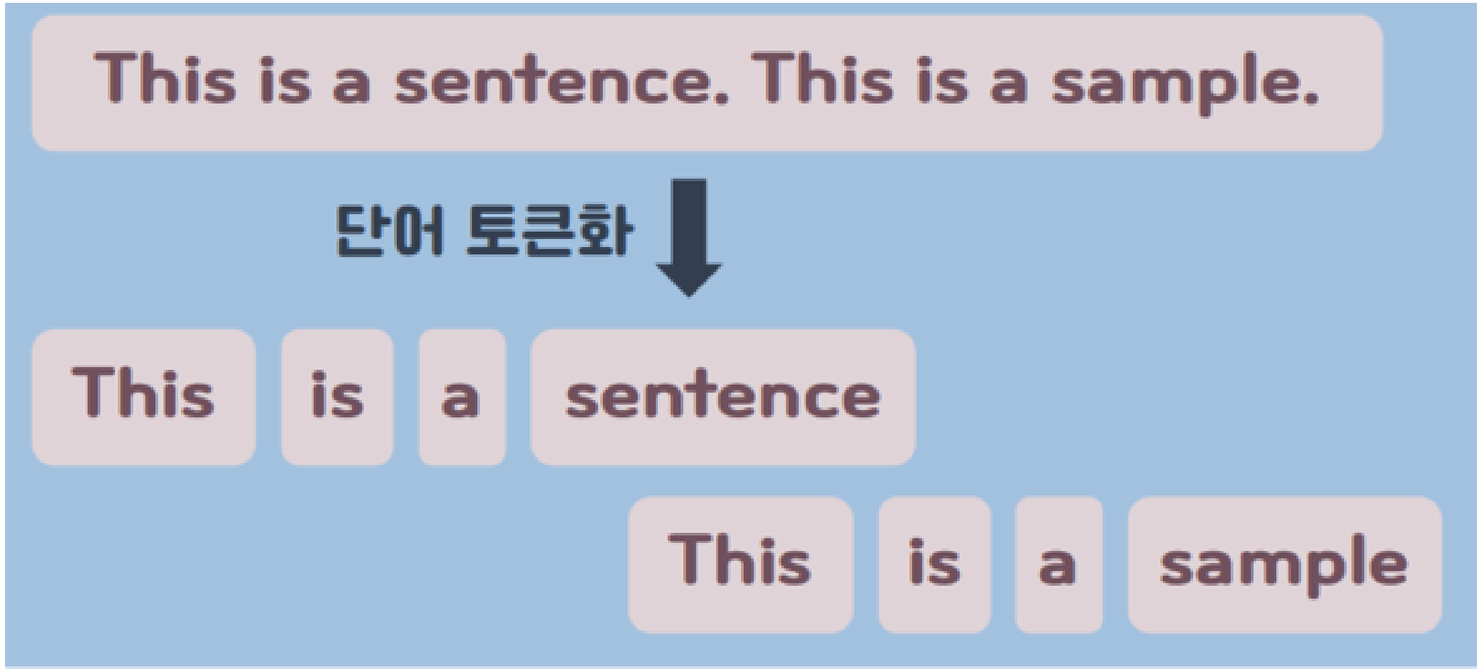
정규화 normalization

표현 방법이 다른 단어들을 통합시켜서 같은 단어로 만들어준다.

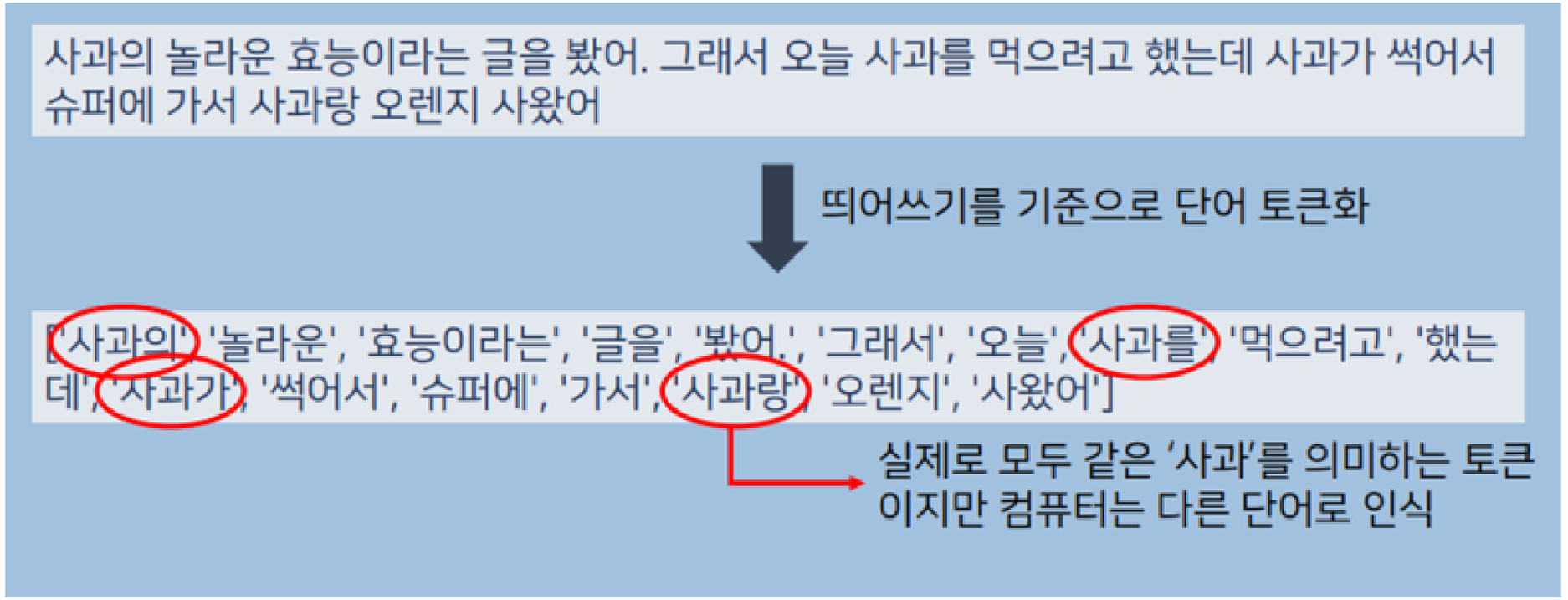
토큰화


This is a sentence. This is a sample.
문장 토큰화
전체 텍스트를 그대로 input하면 컴퓨터는 이 텍스트를 하나의 덩어리로 인식
This is a sentence
This is a sample
텍스트를 일정한 단위로 분리하여 그 내용을 인식할 수 있도록 해주어야 함.

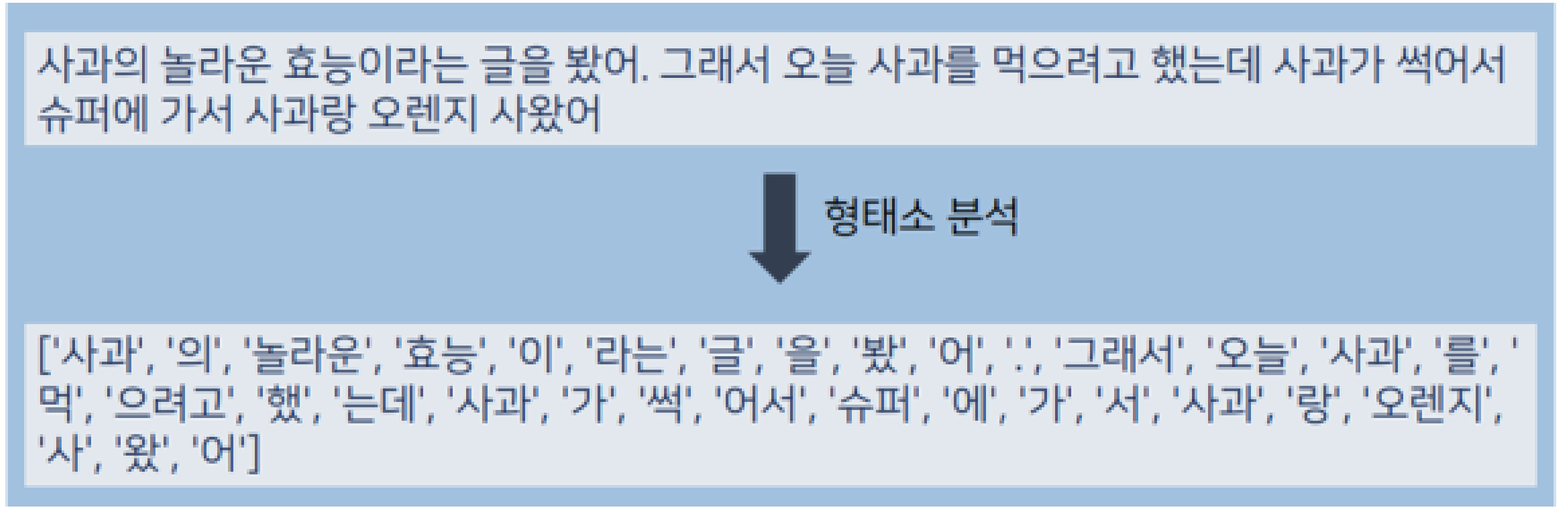
토큰화


This is a sentence. This is a sample.
단어 토큰화
This
is
a
sentence
This
is
a
sample

한국어 토큰화

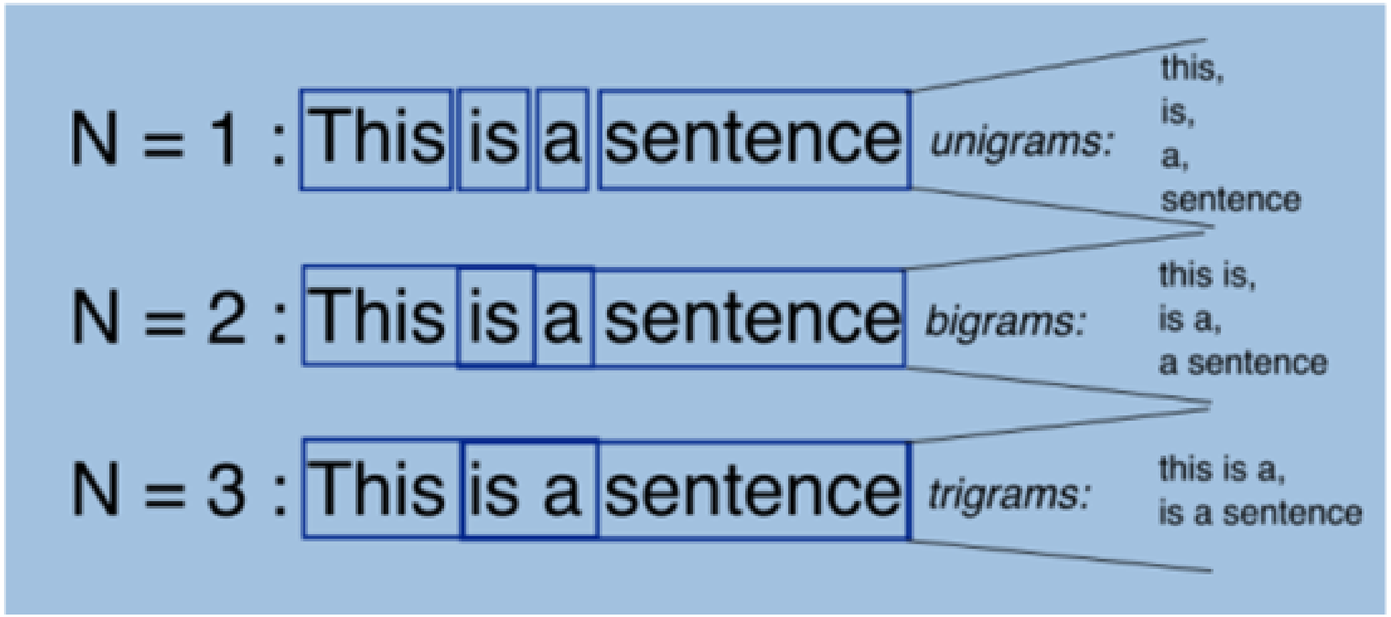

사과의 놀라운 효능이라는 글을 봤어. 그래서 오늘 사과를 먹으려고 했는데 사과가 썩어서 슈퍼에 가서 사과랑 오렌지 사왔어
띄어쓰기를 기준으로 단어 토큰화
['사과의', '놀라운', '효능이라는', '글을', '봤어.', '그래서', '오늘', '사과를', '먹으려고', '했는데', '사과가', '썩어서', '슈퍼에', '가서', '사과랑', '오렌지', '사왔어']
실제로 모두 같은 '사과'를 의미하는 토큰이지만 컴퓨터는 다른 단어로 인식

한국어 토큰화



형태소를 기준으로 문법적 도구와 의미를 가지는 부분을 나누어서 토큰화하기

n-gram 토큰화



$$P(x_1, x_2, x_3, x_4) = P(x_1)P(x_2)P(x_3)P(x_4),$$
$$P(x_1, x_2, x_3, x_4) = P(x_1)P(x_2 \mid x_1)P(x_3 \mid x_2)P(x_4 \mid x_3),$$
$$P(x_1, x_2, x_3, x_4) = P(x_1)P(x_2 \mid x_1)P(x_3 \mid x_1, x_2)P(x_4 \mid x_2, x_3).$$

어간 추출과 표제어 추출

Stemming

문법 요소에 따라 변환된 단어의 원형(어근)을 추출하기 위해 어간을 추출하는 방법.

일정한 규칙을 갖고 단어의 어미를 자르기 때문에 간단하나 섬세하지 않아서 정확성이 떨어짐.

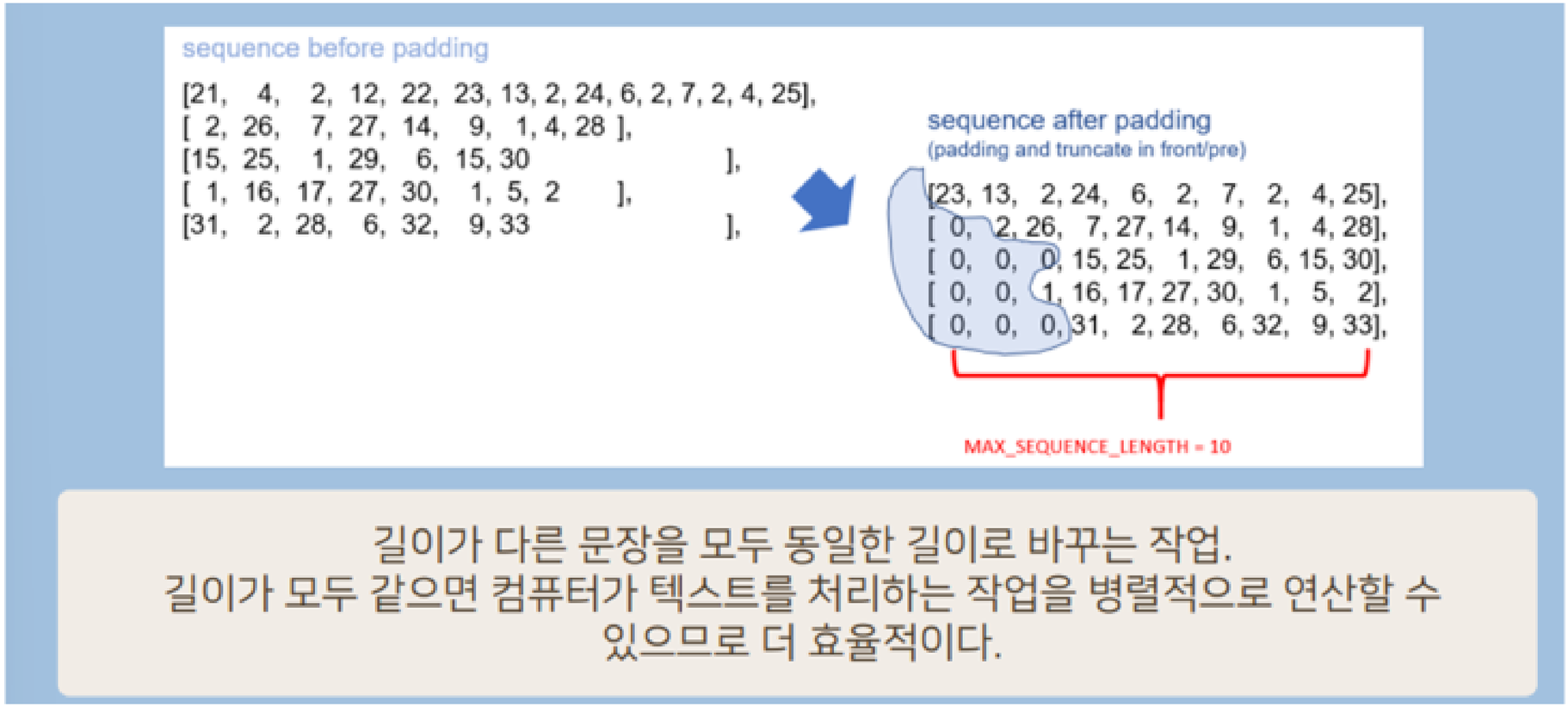
Lemmatization

특정한 규칙을 따르는 것이 아니라 문법 요소와 의미 요소를 감안해서 어근을 추출하는 방법.

Stemming에 비해 높은 정확성을 보이는 대신 시간이 오래 걸림.

ex. 'am', 'are', 'is'를 'be'로 추출

패딩 padding

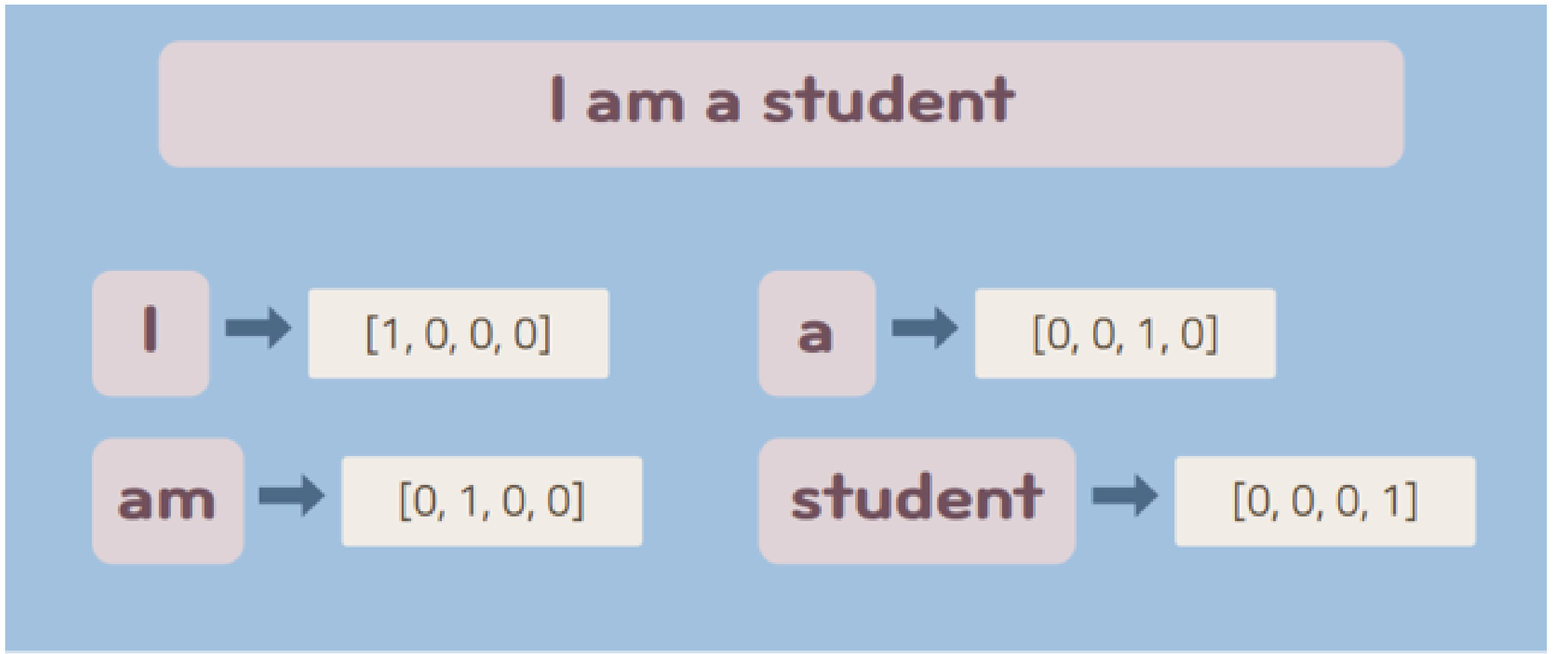


길이가 다른 문장을 모두 동일한 길이로 바꾸는 작업.
길이가 모두 같으면 컴퓨터가 텍스트를 처리하는 작업을 병렬적으로 연산할 수 있으므로 더 효율적이다.

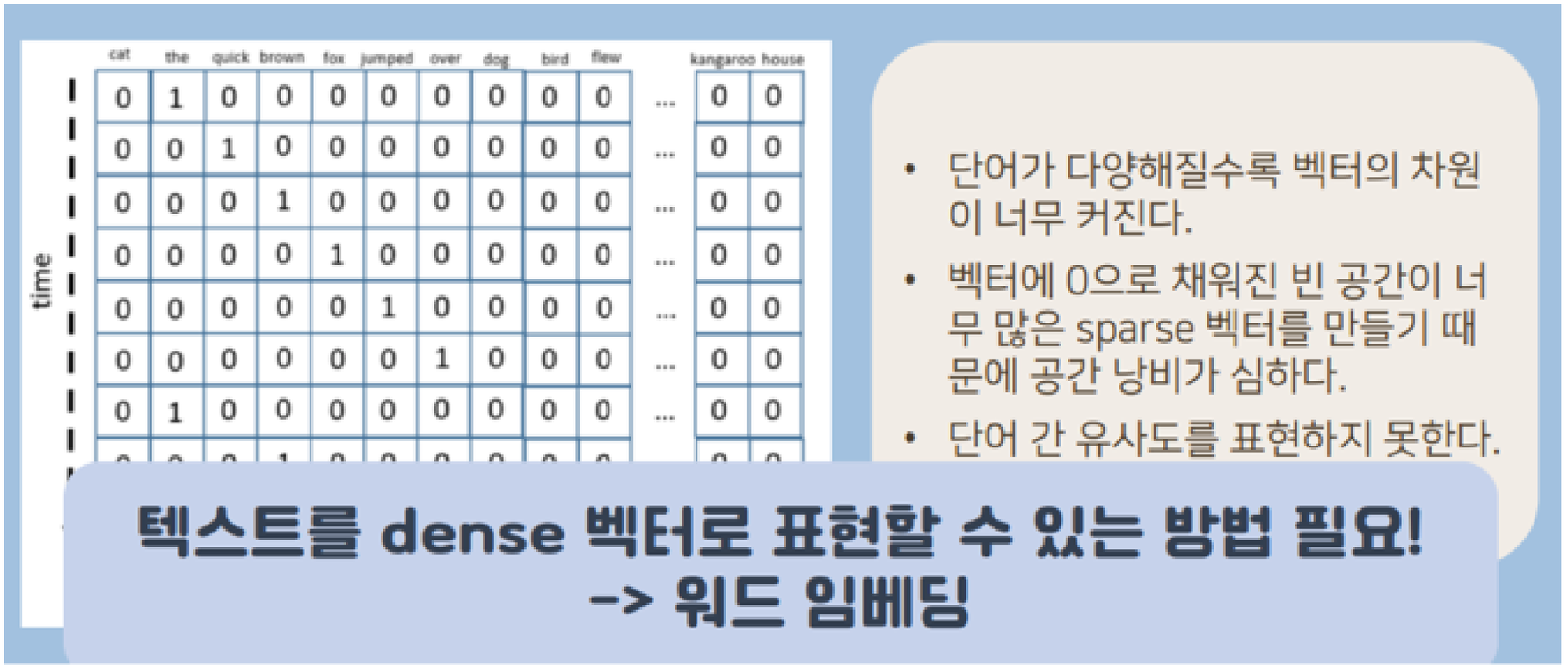
04. embedding

04. embedding

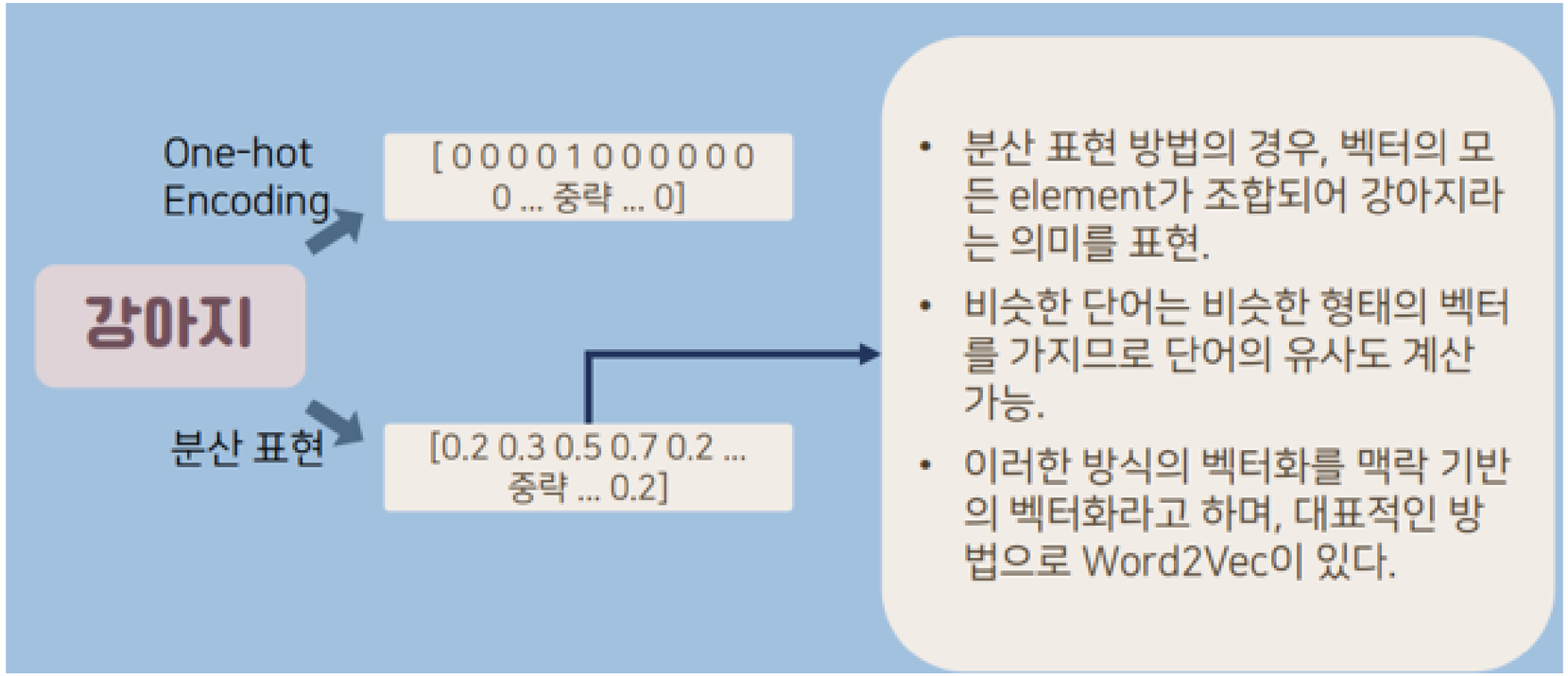
one-hot encoding



one-hot encoding의 문제점

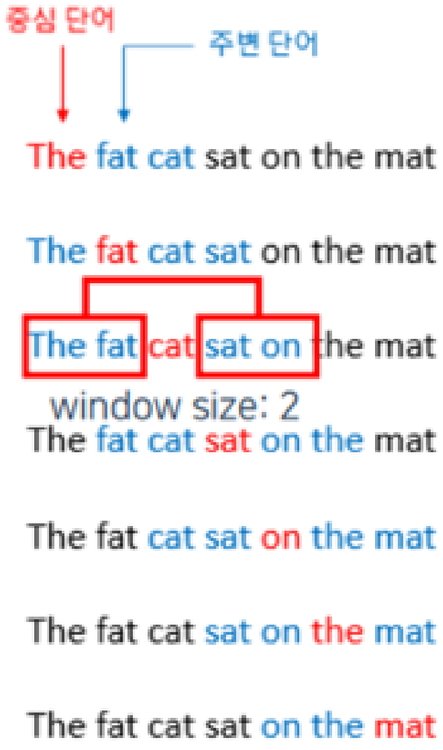


예측 기반 임베딩


One-hot Encoding
[0 0 0 0 0 1 0 0 0 0 0 0 0 ... 중략 ... 0]
강아지
분산 표현
[0.2 0.3 0.5 0.7 0.2 ... 중략 ... 0.2]
• 분산 표현 방법의 경우, 벡터의 모든 element가 조합되어 강아지라는 의미를 표현.
• 비슷한 단어는 비슷한 형태의 벡터를 가지므로 단어의 유사도 계산 가능.
• 이러한 방식의 벡터화를 맥락 기반의 벡터화라고 하며, 대표적인 방법으로 Word2Vec이 있다.

04. embedding

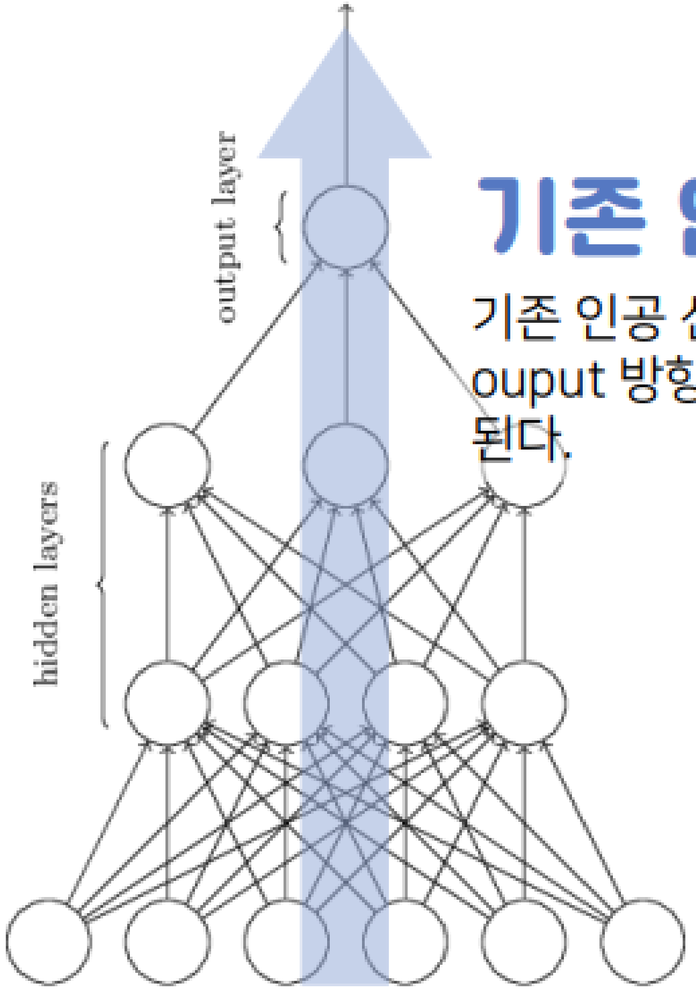
word2vec



중심 단어	주변 단어
[1, 0, 0, 0, 0, 0, 0]	[0, 1, 0, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0, 0]
[0, 1, 0, 0, 0, 0, 0]	[1, 0, 0, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0, 0], [0, 0, 0, 1, 0, 0, 0]
[0, 0, 1, 0, 0, 0, 0]	[1, 0, 0, 0, 0, 0, 0], [0, 1, 0, 0, 0, 0, 0], [0, 0, 0, 1, 0, 0, 0], [0, 0, 0, 0, 1, 0, 0]
[0, 0, 0, 1, 0, 0, 0]	[0, 1, 0, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0, 0], [0, 0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 0, 1, 0]
[0, 0, 0, 0, 1, 0, 0]	[0, 0, 1, 0, 0, 0, 0], [0, 0, 0, 1, 0, 0, 0], [0, 0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 0, 0, 1]
[0, 0, 0, 0, 0, 1, 0]	[0, 0, 0, 1, 0, 0, 0], [0, 0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 0, 0, 1]
[0, 0, 0, 0, 0, 0, 1]	[0, 0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 0, 1, 0]

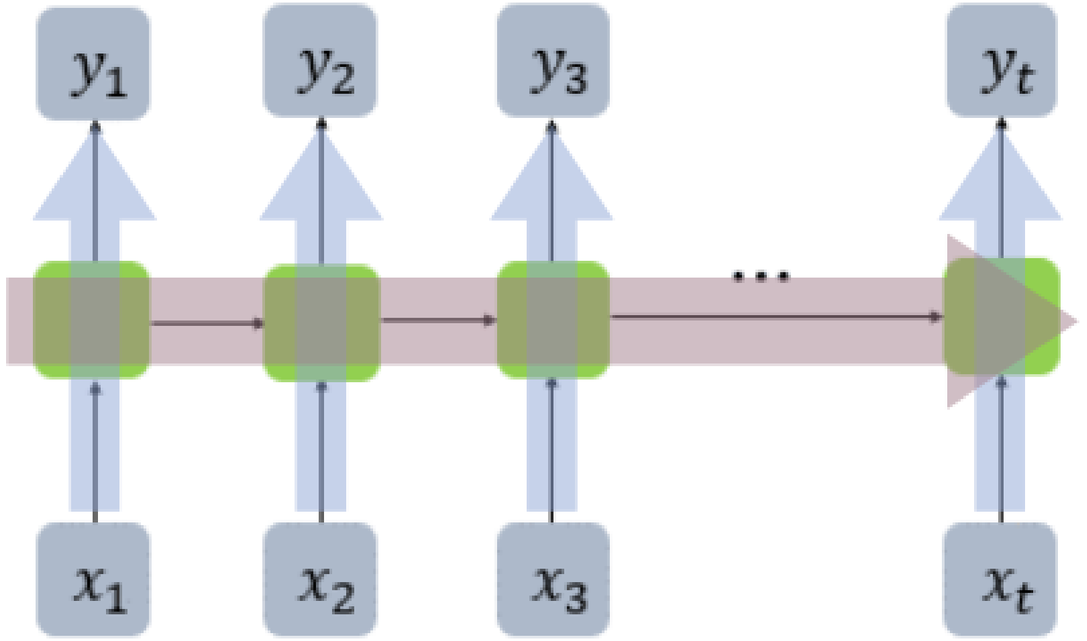
05. RNN

05. RNN



기존 인공 신경망

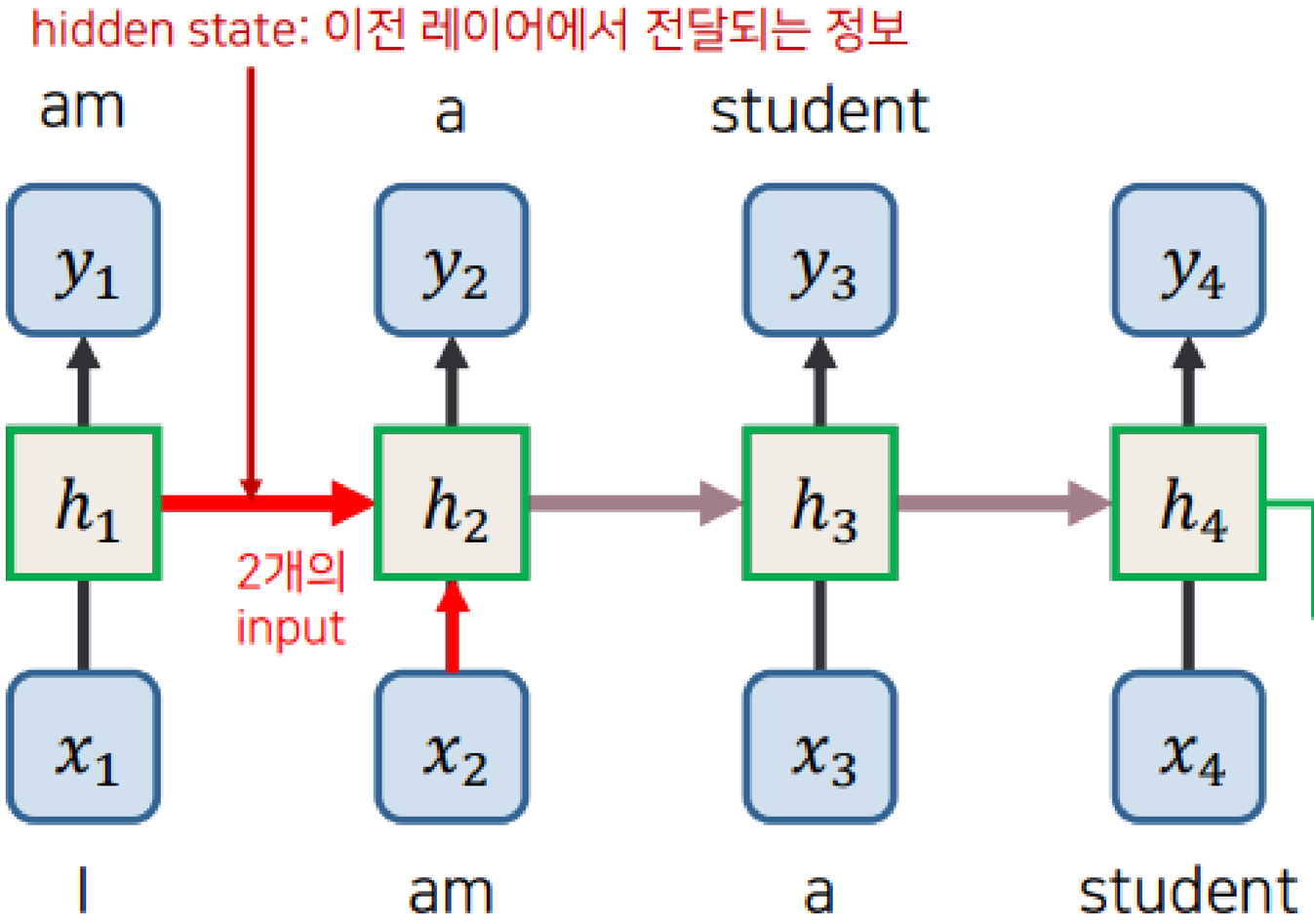
기존 인공 신경망은 input에서 ouput 방향으로만 학습이 진행된다.



RNN (순환 신경망)

학습이 한 방향으로 진행되지 않는다.
Input->output 방향 이외에 다음 time step의 hidden layer 방향으로도 학습이 진행된다.

05. RNN

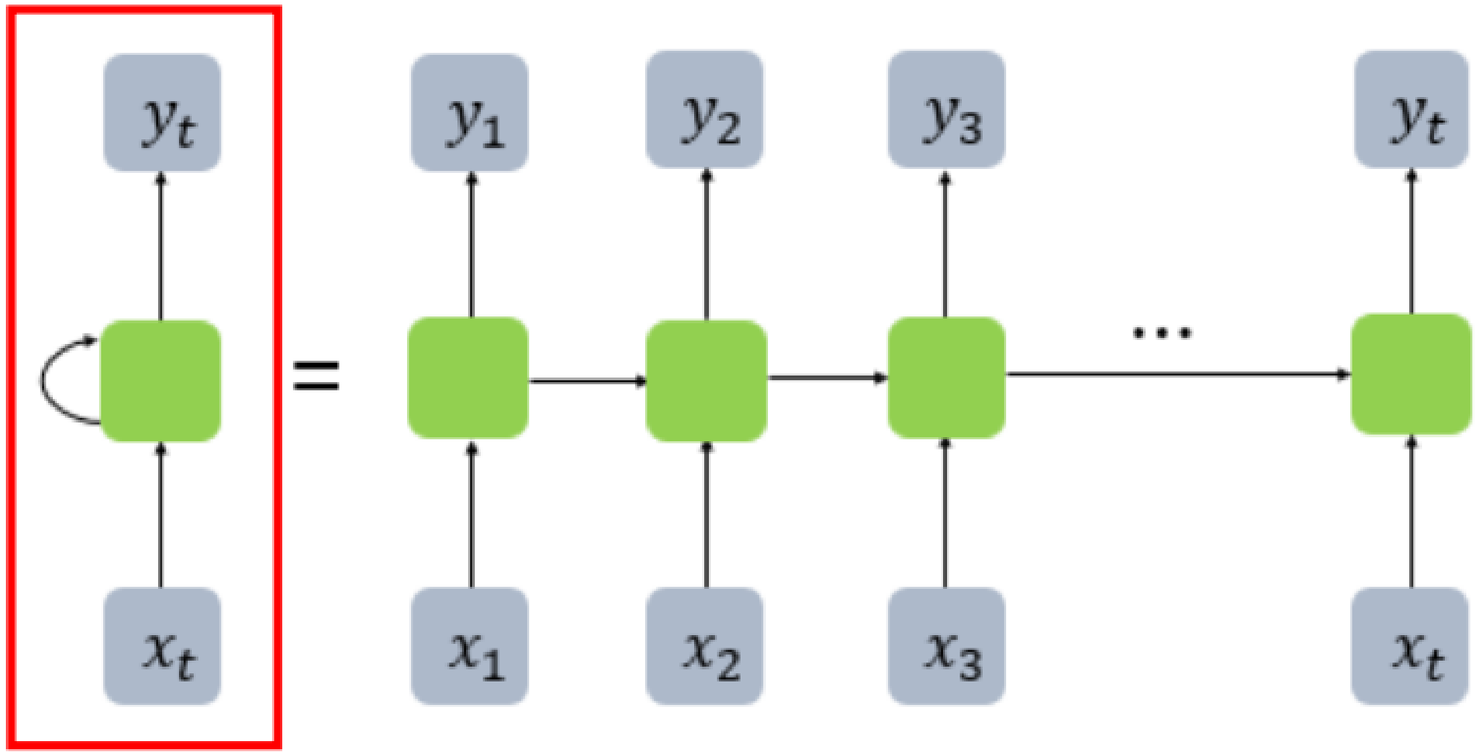


재귀적 학습: t-1번째 hidden layer가 t번째 hidden layer로 전달되는 학습.

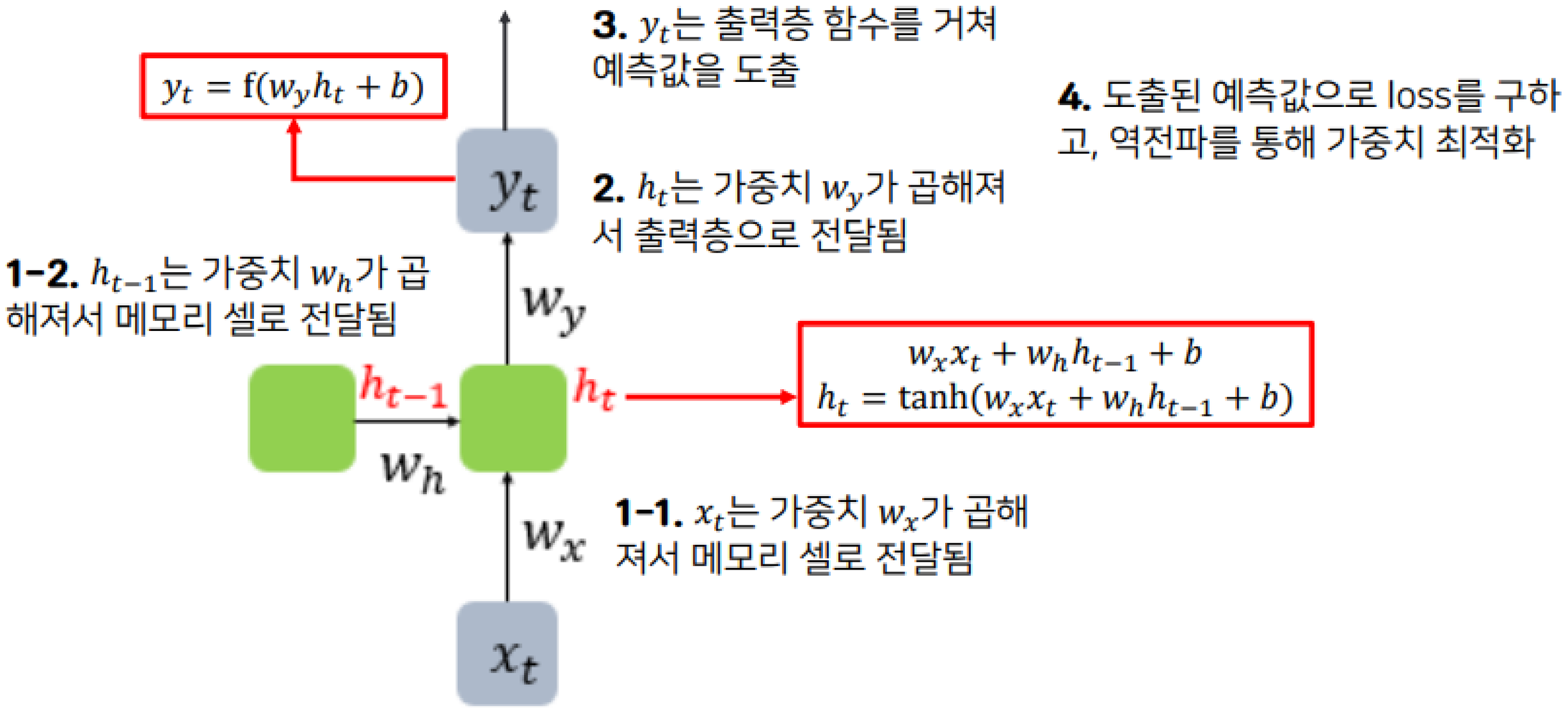
RNN은 해당 time step의 input+이전 time step hidden layer까지 포함해서 학습

메모리 셀: 이전 값을 기억하려는 특징을 가짐

05. RNN



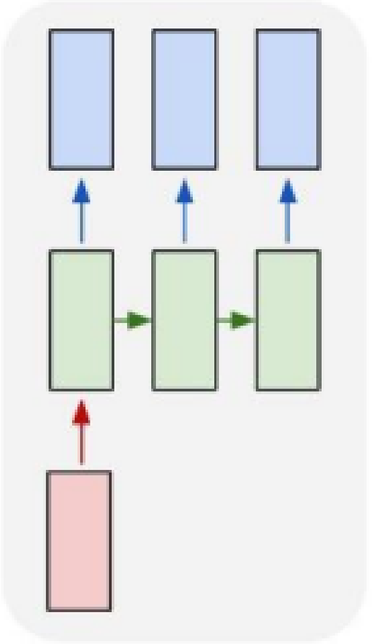
05. RNN


$y_t = \mathrm{f}(w_y h_t + b)$
3. y_t는 출력층 함수를 거쳐 예측값을 도출
4. 도출된 예측값으로 loss를 구하고, 역전파를 통해 가중치 최적화
y_t
2. h_t는 가중치 w_y가 곱해져서 출력층으로 전달됨
1-2. h_{t-1}는 가중치 w_h가 곱해져서 메모리 셀로 전달됨
w_y
h_{t-1}
h_t
$w_x h_{t-1} + b$
$h_t = \tanh(w_x x_t + w_h h_{t-1} + b)$
w_h
w_x
1-1. x_t는 가중치 w_x가 곱해져서 메모리 셀로 전달됨
x_t

05. RNN

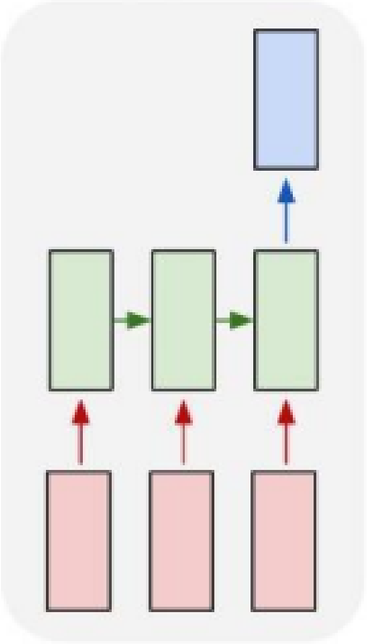
RNN은 입력층과 출력층 길이를 다르게 설정하여 다양한 구조를 만들 수 있다.

one to many



한 개의 input이 여러 개의 output(sequence)을 출력하는 구조.
image captioning 등에 사용할 수 있다.

many to one

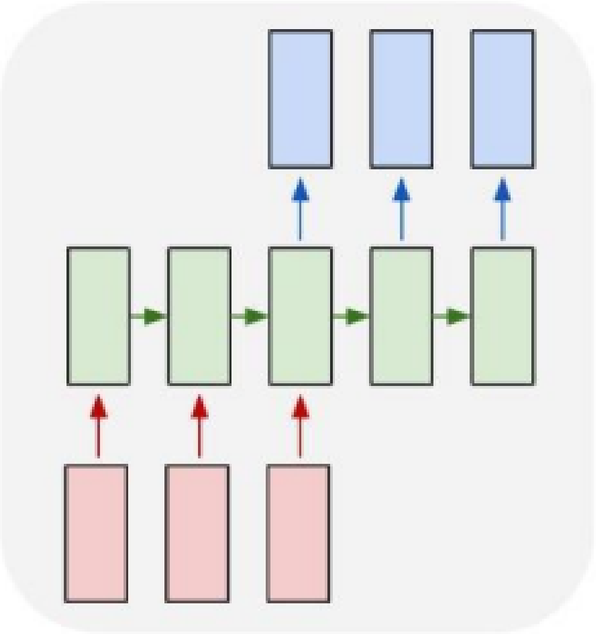


여러 개의 input(sequence)이 한 개의 output을 출력하는 구조.
감성 분석 등에 사용할 수 있다.

05. RNN

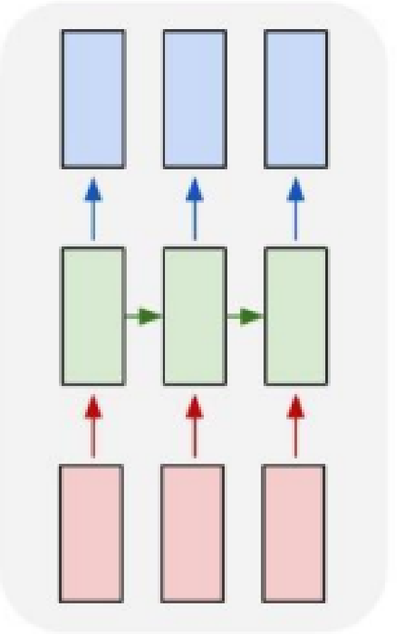
RNN은 입력층과 출력층 길이를 다르게 설정하여 다양한 구조를 만들 수 있다.

many to many



여러 개의 input(sequence)이 여러 개의 output(sequence)을 출력하는 구조.
기계 번역 등에 사용할 수 있다.

many to many



many-to-many 모델 중에서도 각 input마다 output이 생성되는 모델.
출력이 지연되지 않기 때문에 실시간 처리가 필요한 작업에 사용할 수 있다.

06.

Week2
과제

THANK YOU

YOUR NAME