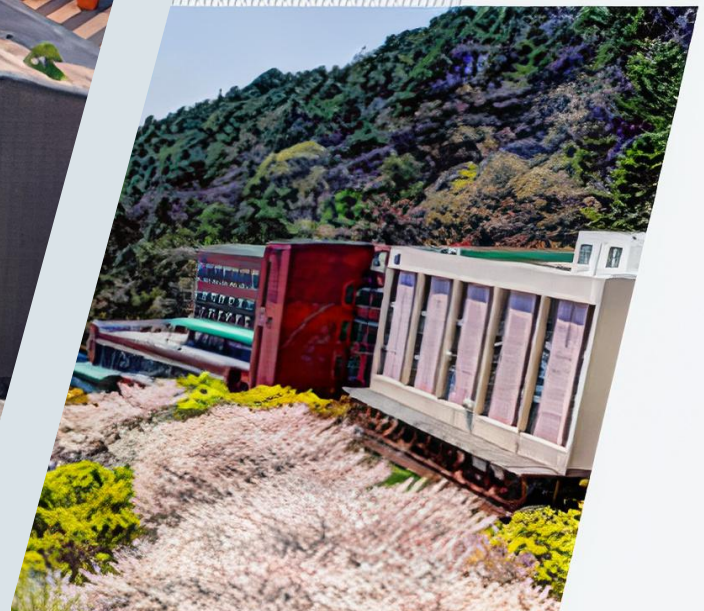
Dong-A Univ. (ISPL)

동아대학교 DONG-A UNIVERSITY

[Language Model] Word2Vec 이론



컴퓨터AI공학부

2025년 1학기 머신러닝

Large Language Model (LLM)

- **Language Model**
  - 인간의 언어를 이해하고 생성하기 위해 설계된 인공지능 모델
- **Large Language Model**
  - 방대한 양의 데이터에 기반하여 Training된 수많은 Parameter를 가지는 인공지능 모델

Large Language Model (LLM)

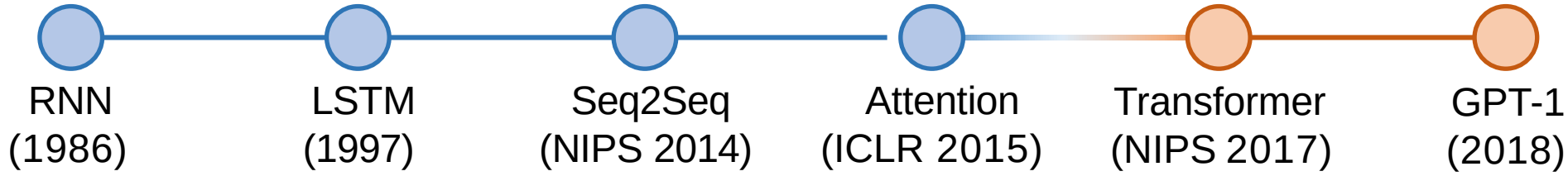
- **Language Model**
  - 인간의 언어를 이해하고 생성하기 위해 설계된 인공지능 모델
- **Large Language Model**
  - 방대한 양의 데이터에 기반하여 Training된 수많은 Parameter를 가지는 인공지능 모델
  - 대표적인 LLM: **Chat GPT**

ChatGPT

Generative **P**re-trained **T**ransformer

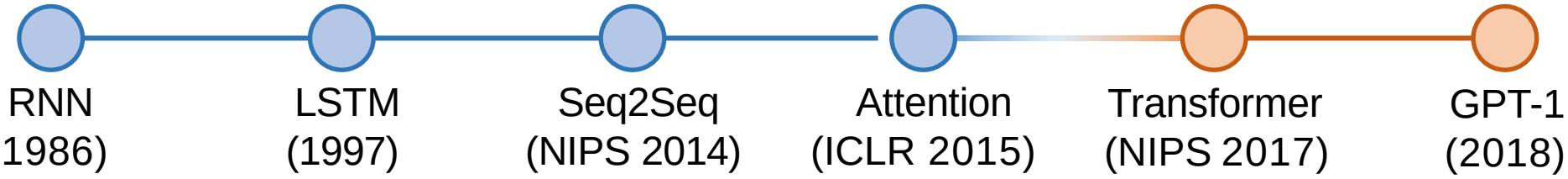
Large Language Model (LLM)

- **History of Language Model**
  - 기존 Language model은 RNN 기반으로 설계 되었음
  - RNN 기반 모델들의 문제점을 Transformer가 해결하면서 높은 성능을 도출함



Large Language Model (LLM)

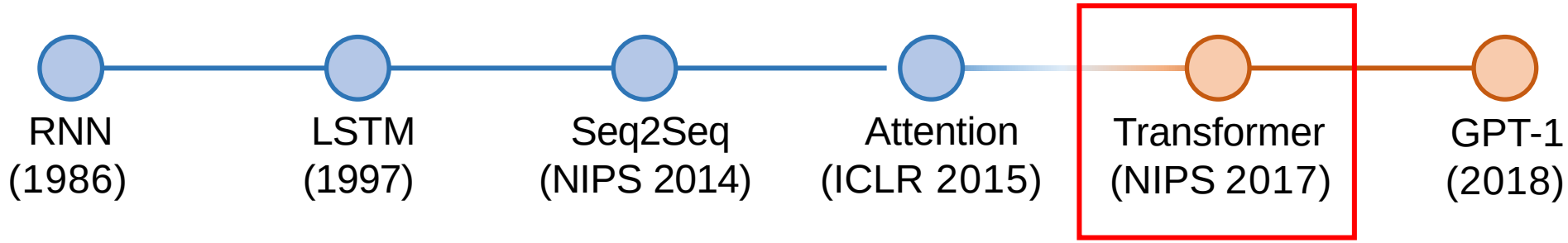
- **History of Language Model**
  - 기존 Language model은 RNN 기반으로 설계 되었음
  - RNN 기반 모델들의 문제점을 Transformer가 해결하면서 높은 성능을 도출함



RNN을 기반으로 설계된 model

Large Language Model (LLM)

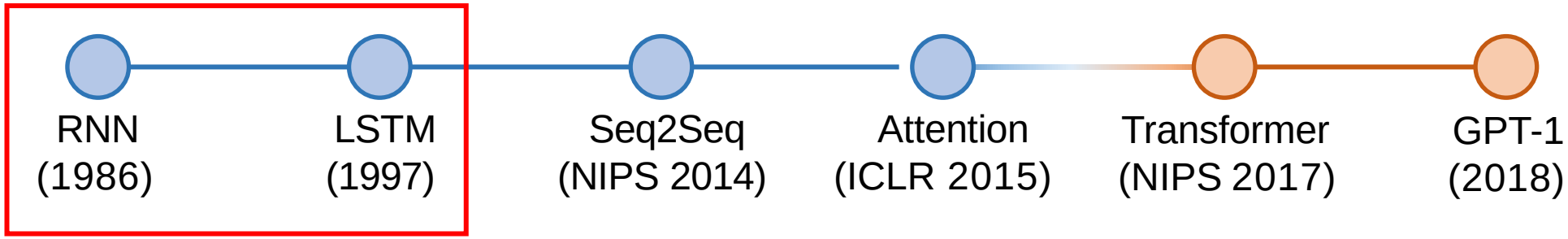
- **History of Language Model**
  - 기존 Language model은 RNN 기반으로 설계 되었음
  - RNN 기반 모델들의 문제점을 Transformer가 해결하면서 높은 성능을 도출함



RNN 기반 Model의 한계점 해결

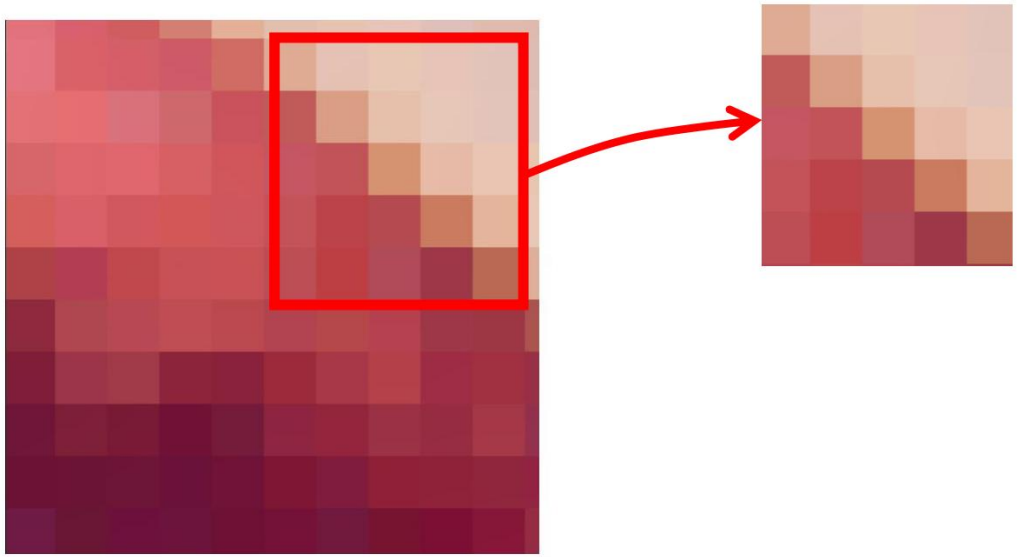
Large Language Model (LLM)

- **History of Language Model**
  - 기존 Language model은 RNN 기반으로 설계 되었음
  - RNN 기반 모델들의 문제점을 Transformer가 해결하면서 높은 성능을 도출함



RNN

- **Overview**
  - 기존의 CNN은 데이터의 공간적인 정보만을 학습함



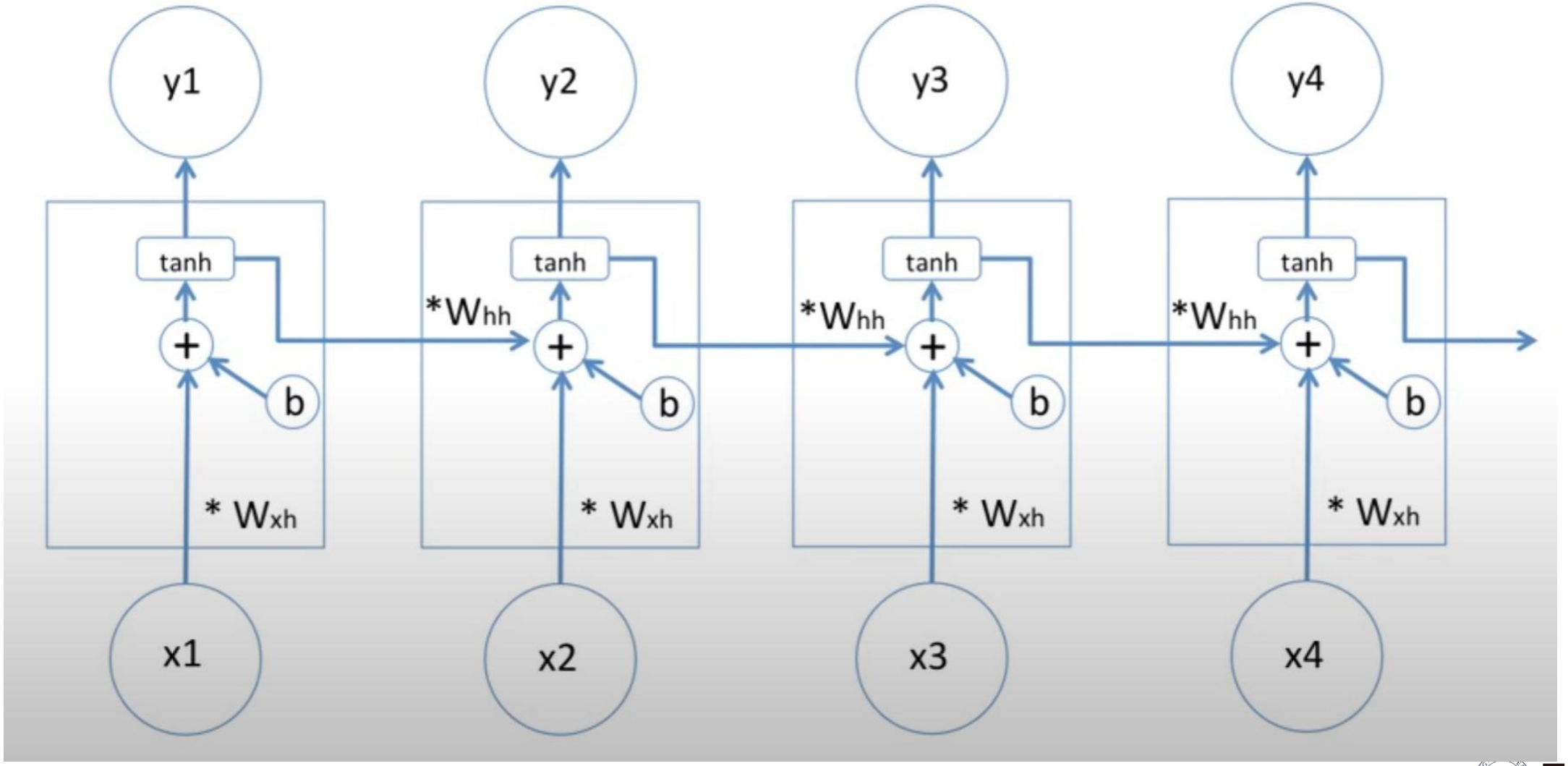
CNN은 이미지 데이터의 공간적 특징을 추출하여 학습

Review (RNN & LSTM)

RNN

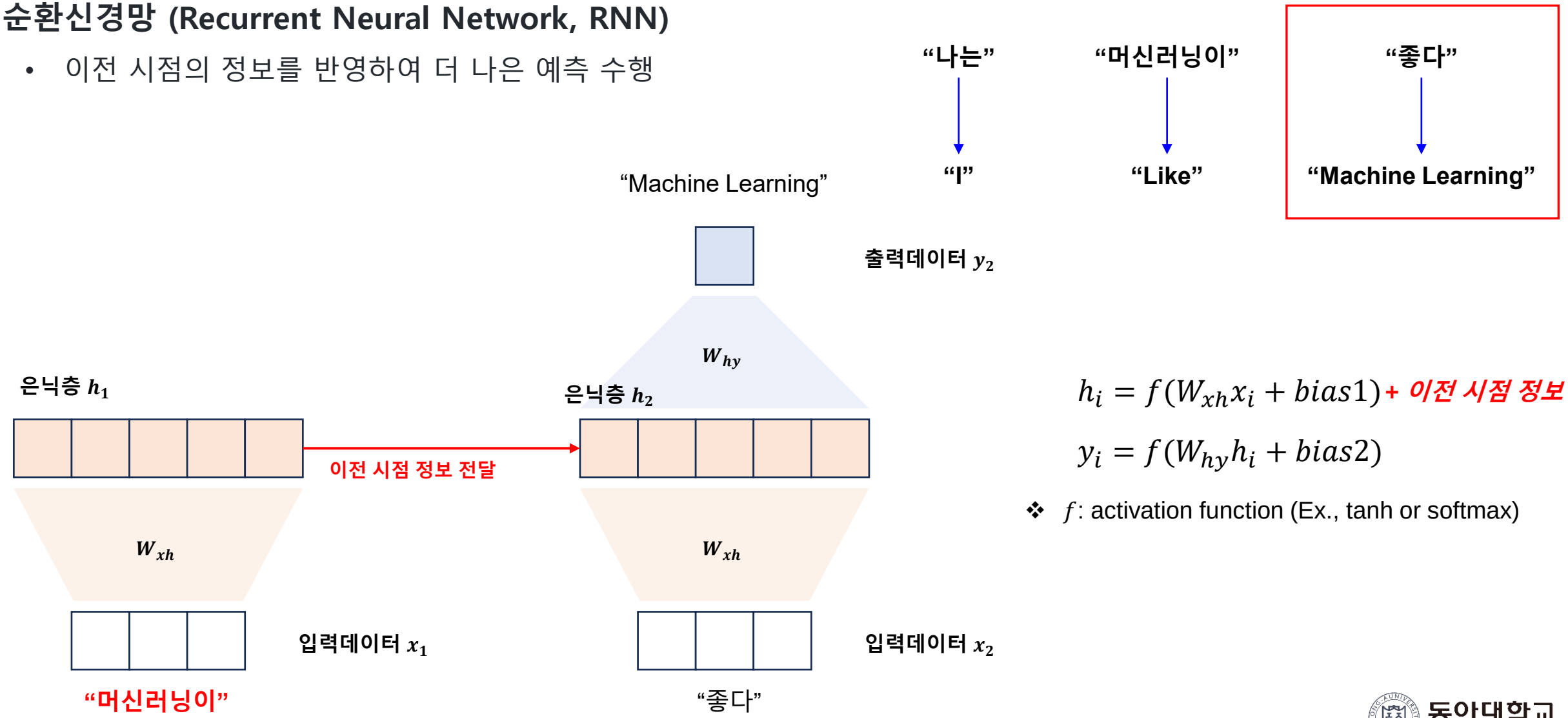
$$h_t = \tanh(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$$
$$y_t = W_{hy}h_t + b_y$$

- 여기서 $W_{xh}, W_{hh}, b_h, W_{hy}, b_y$는 **모두 time-step 간 공유됨**



RNN

- **순환신경망 (Recurrent Neural Network, RNN)**
  - 이전 시점의 정보를 반영하여 더 나은 예측 수행



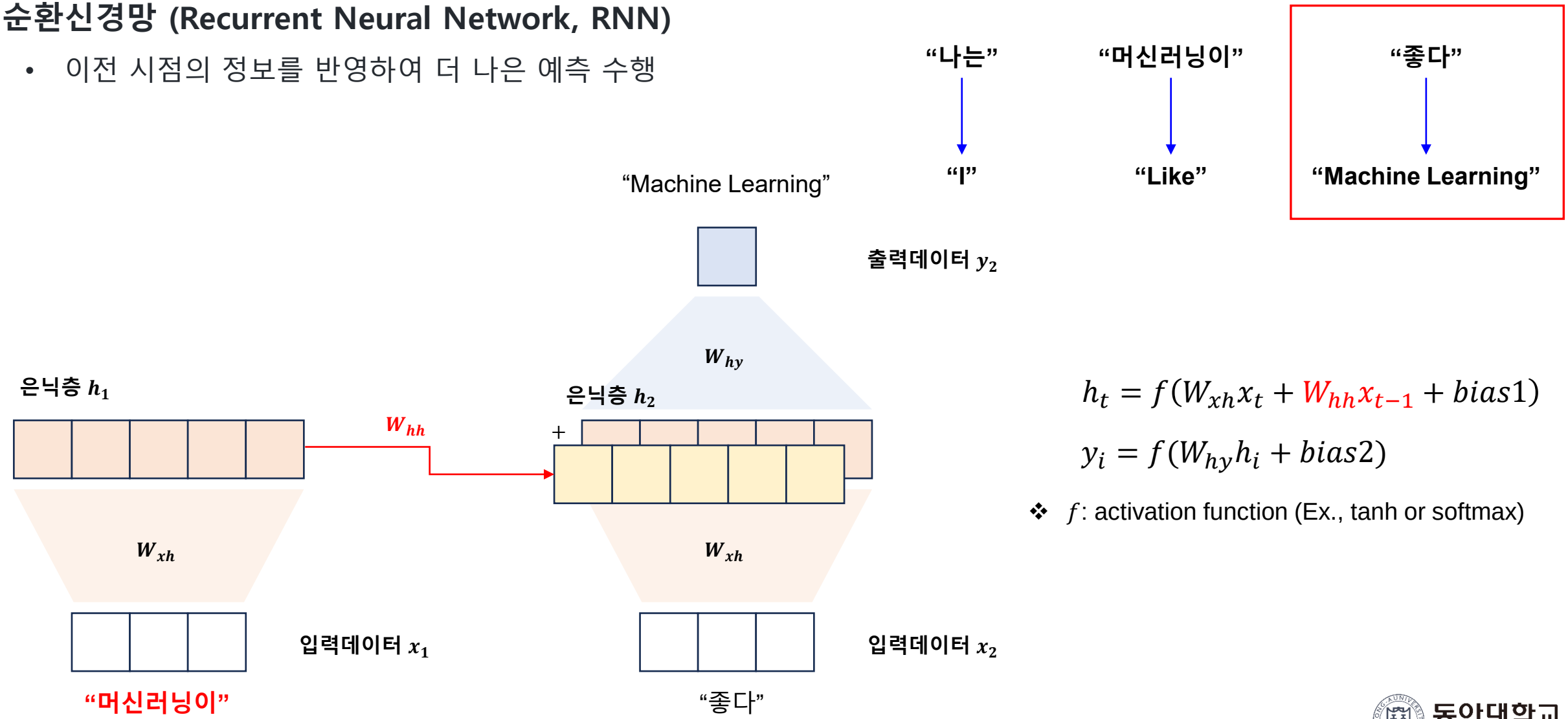
$$h_i = f(W_{xh}x_i + bias1) \text{ + 이전 시점 정보}$$

$$y_i = f(W_{hy}h_i + bias2)$$

❖ f: activation function (Ex., tanh or softmax)

RNN

- **순환신경망 (Recurrent Neural Network, RNN)**
  - 이전 시점의 정보를 반영하여 더 나은 예측 수행



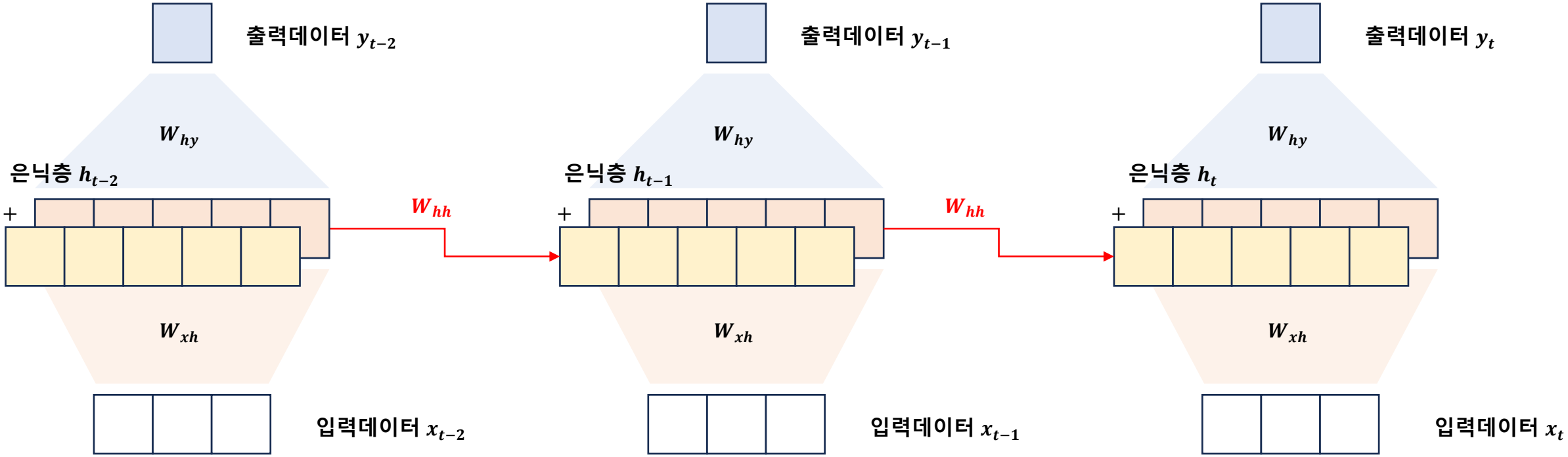
$$h_t = f(W_{xh}x_t + W_{hh}x_{t-1} + bias1)$$

$$y_i = f(W_{hy}h_i + bias2)$$

❖ f: activation function (Ex., tanh or softmax)

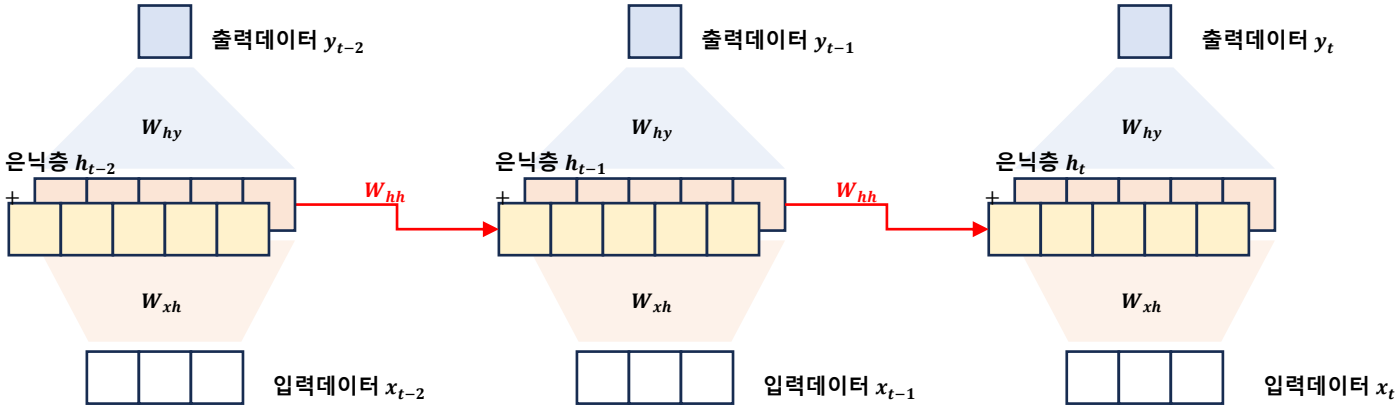
RNN

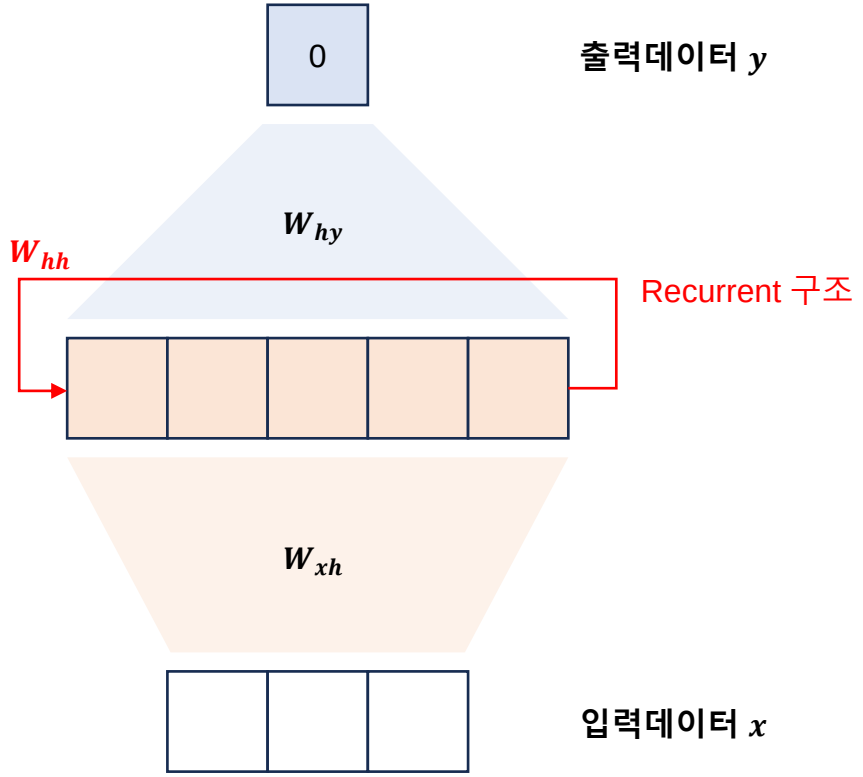
- **순환신경망 (Recurrent Neural Network, RNN)**
  - 이전 시점의 정보를 반영하여 더 나은 예측 수행
  - 이전 정보들이 순환 (반복)하여 입력



RNN

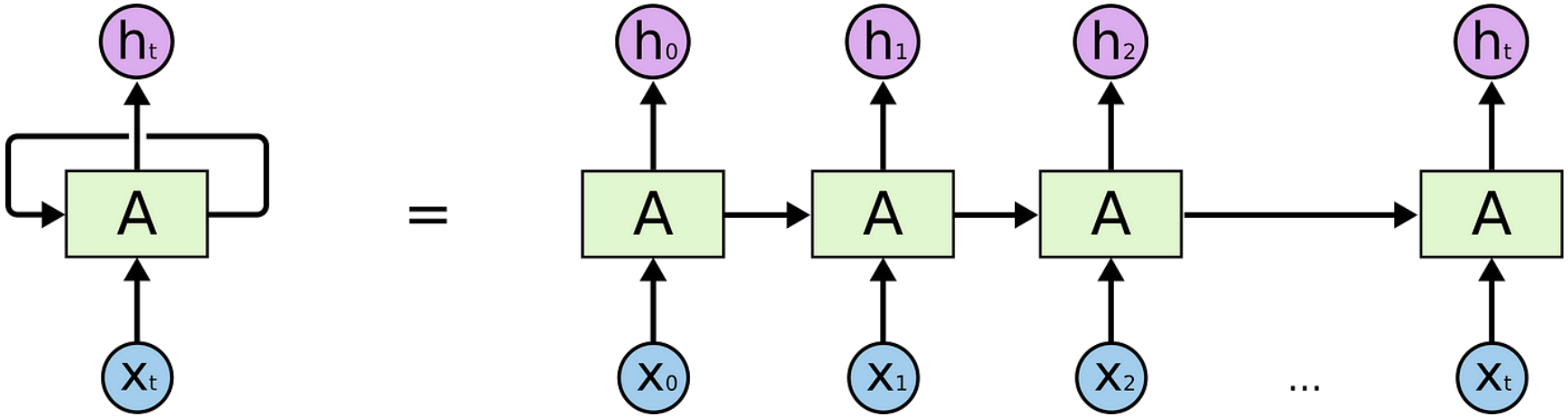
- **순환신경망 (Recurrent Neural Network, RNN)**
  - 이전 시점의 정보를 반영하여 더 나은 예측 수행
  - 이전 정보들이 순환 (반복)하여 입력





RNN

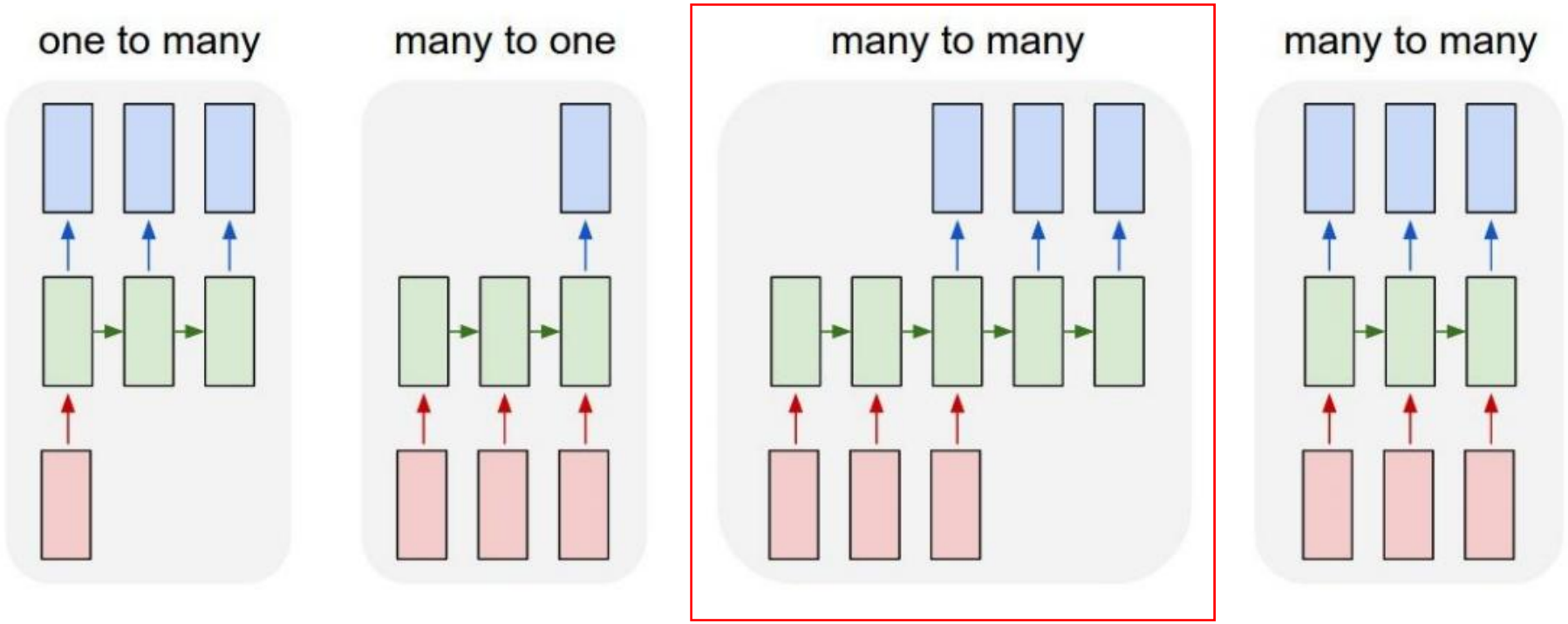
- **Overview**
  - 시간 순서가 있는 데이터 (Time Series Data)를 효율적으로 학습하기 위해 등장
  - 순환 구조를 통해서 이전 상태의 정보를 함께 사용하여 현재 상태의 정보 학습



순서가 있는 데이터를 순차적으로 입력하여 학습

RNN

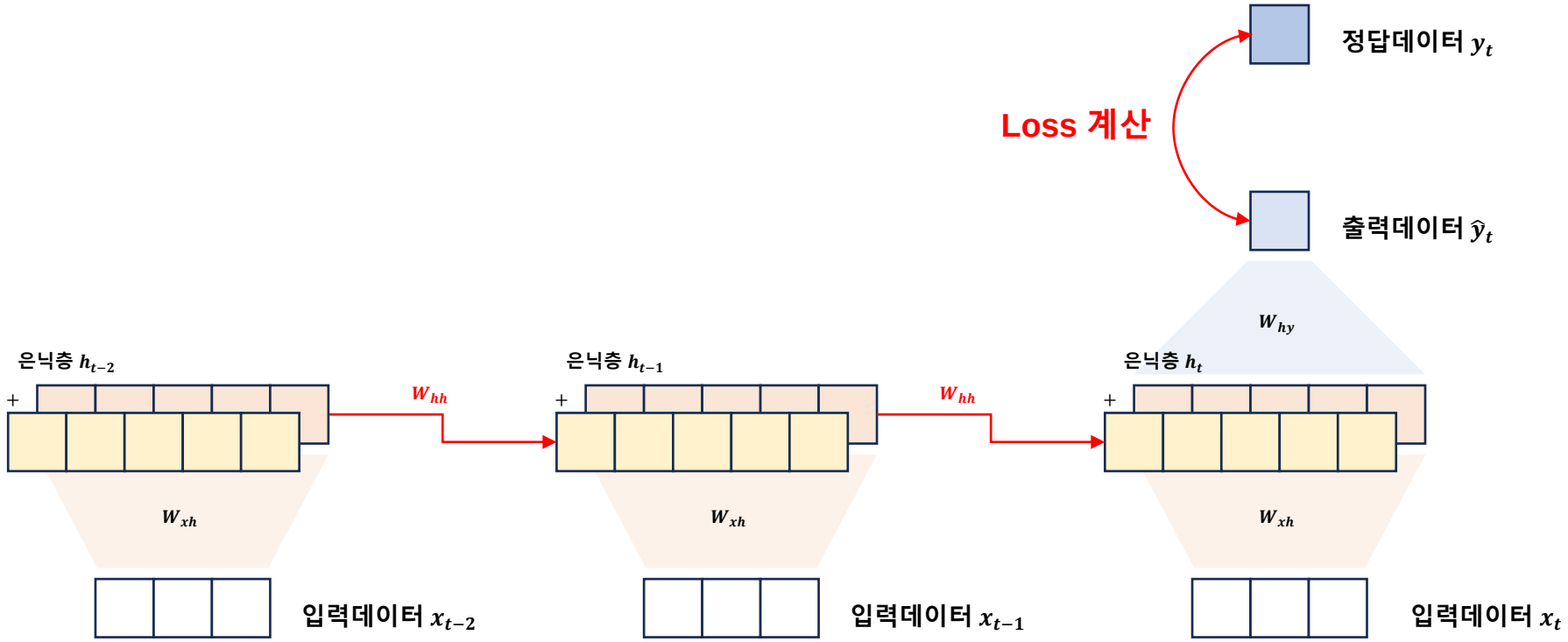
- **순환신경망 구조의 종류**
  - 순차적인 입력의 길이, 순차적인 예측의 길이에 따라 다음과 같이 구분 가능



Sequence to Sequence

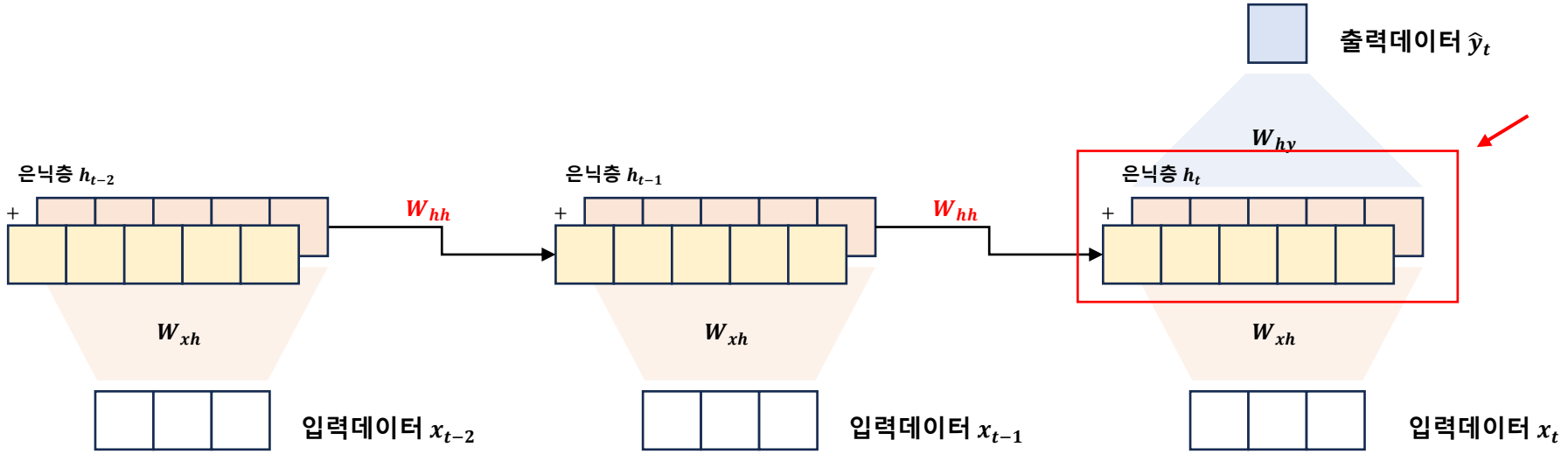
RNN 학습

- **학습 파라미터 (W_{hy}, W_{hh}, W_{xh})**
  - 각 파라미터는 매 시점마다 동일한 값을 사용함 (Shared parameter)



RNN 학습

- **학습 파라미터 (W_{hy}, W_{hh}, W_{xh})**
  - 각 파라미터는 매 시점마다 동일한 값을 사용함 (Shared parameter)

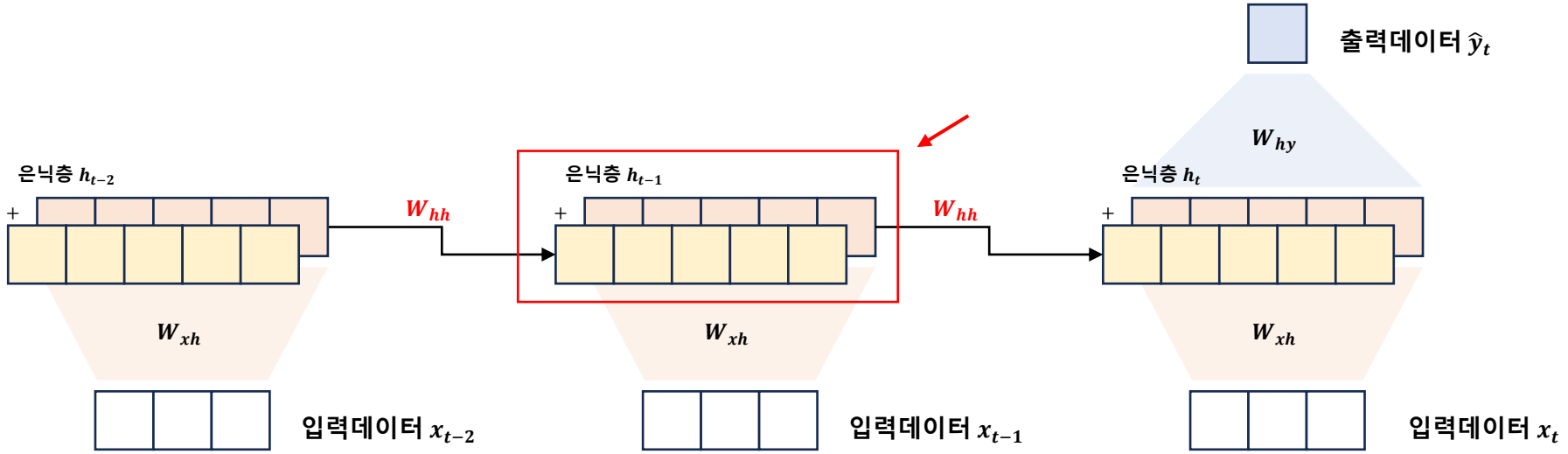


$$\frac{\partial Loss}{\partial W_{hh}} = \frac{\partial Loss}{\partial \hat{y}_t} \times \frac{\partial \hat{y}_t}{\partial h_t} \times \frac{\partial h_t}{\partial W_{hh}} + \frac{\partial Loss}{\partial \hat{y}_t} \times \frac{\partial \hat{y}_t}{\partial h_t} \times \frac{\partial h_t}{\partial h_{t-1}} \times \frac{\partial h_{t-1}}{\partial W_{hh}} + \frac{\partial Loss}{\partial \hat{y}_t} \times \frac{\partial \hat{y}_t}{\partial h_t} \times \frac{\partial h_t}{\partial h_{t-1}} \times \frac{\partial h_{t-1}}{\partial h_{t-2}} \times \frac{\partial h_{t-2}}{\partial W_{hh}}$$

t 시점에서의 영향

RNN 학습

- **학습 파라미터 (W_{hy}, W_{hh}, W_{xh})**
  - 각 파라미터는 매 시점마다 동일한 값을 사용함 (Shared parameter)

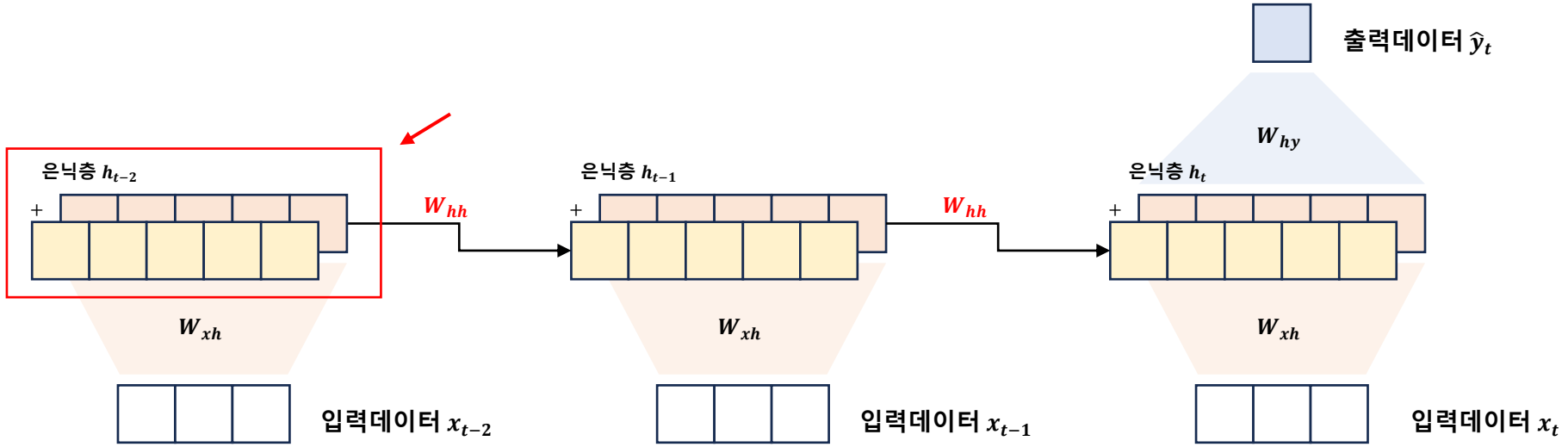


$$\frac{\partial Loss}{\partial W_{hh}} = \frac{\partial Loss}{\partial \hat{y}_t} \times \frac{\partial \hat{y}_t}{\partial h_t} \times \frac{\partial h_t}{\partial W_{hh}} + \frac{\partial Loss}{\partial \hat{y}_t} \times \frac{\partial \hat{y}_t}{\partial h_t} \times \frac{\partial h_t}{\partial h_{t-1}} \times \frac{\partial h_{t-1}}{\partial W_{hh}} + \frac{\partial Loss}{\partial \hat{y}_t} \times \frac{\partial \hat{y}_t}{\partial h_t} \times \frac{\partial h_t}{\partial h_{t-1}} \times \frac{\partial h_{t-1}}{\partial h_{t-2}} \times \frac{\partial h_{t-2}}{\partial W_{hh}}$$

t -1 시점에서의 영향

RNN 학습

- **학습 파라미터 (W_{hy}, W_{hh}, W_{xh})**
  - 각 파라미터는 매 시점마다 동일한 값을 사용함 (Shared parameter)

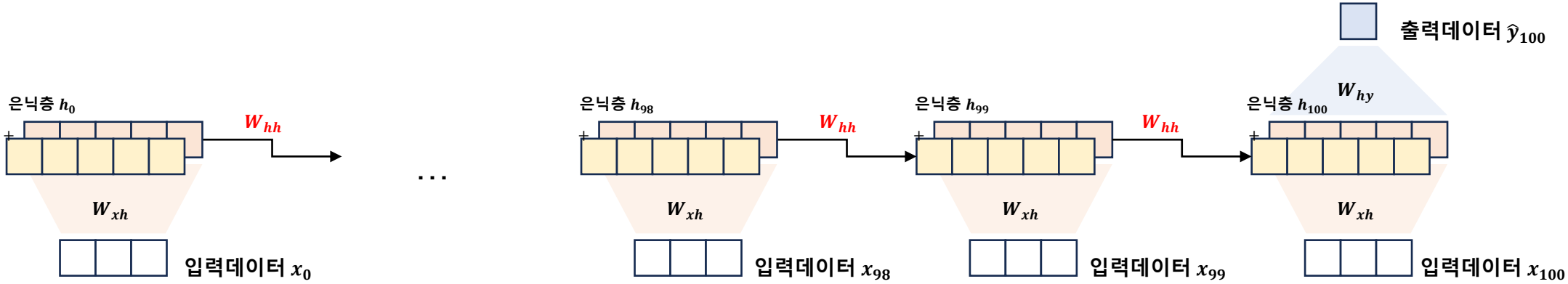


$$\frac{\partial Loss}{\partial W_{hh}} = \frac{\partial Loss}{\partial \hat{y}_t} \times \frac{\partial \hat{y}_t}{\partial h_t} \times \frac{\partial h_t}{\partial W_{hh}} + \frac{\partial Loss}{\partial \hat{y}_t} \times \frac{\partial \hat{y}_t}{\partial h_t} \times \frac{\partial h_t}{\partial h_{t-1}} \times \frac{\partial h_{t-1}}{\partial W_{hh}} + \frac{\partial Loss}{\partial \hat{y}_t} \times \frac{\partial \hat{y}_t}{\partial h_t} \times \frac{\partial h_t}{\partial h_{t-1}} \times \frac{\partial h_{t-1}}{\partial h_{t-2}} \times \frac{\partial h_{t-2}}{\partial W_{hh}}$$

t -2 시점에서의 영향

RNN의 한계점

- **장기 의존성 문제 (Long-term dependency problem)**
  - Sequence의 길이가 길어질수록, 과거 정보 학습에 어려움이 발생함
  - 학습 과정 중 기울기 소실 (Vanishing Gradient) 발생



$$\frac{\partial Loss}{\partial W_{hh}} = \ldots + \frac{\partial Loss}{\partial \hat{y}_{100}} \times \frac{\partial \hat{y}_{100}}{\partial h_{100}} \times \frac{\partial h_{100}}{\partial h_{99}} \times \frac{\partial h_{99}}{\partial h_{98}} \times \frac{\partial h_{98}}{\partial h_{97}} \times \frac{\partial h_{97}}{\partial h_{96}} \times \ldots \times \frac{\partial h_3}{\partial h_2} \times \frac{\partial h_2}{\partial h_1} \times \frac{\partial h_1}{\partial h_0} \times \frac{\partial h_0}{\partial W_{hh}} \approx 0$$

0 ~ 1 0 ~ 1 0 ~ 1 0 ~ 1 0 ~ 1 0 ~ 1 0 ~ 1 0 ~ 1

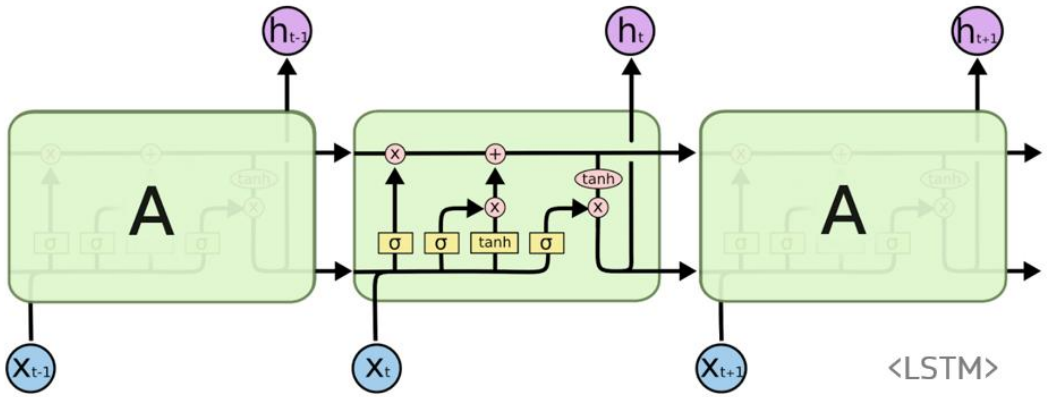
→ 기울기가 소실되어 parameter가 업데이트 되지 않음

t = 0 시점에서의 영향

Review (RNN & LSTM)

LSTM

- **장단기 메모리 순환신경망 (Long Short-Term Memory)**
  - RNN의 장기 의존성 문제를 완화한 개선 모델
  - Cell state 구조를 제안하고 세가지 gate 추가함
    - Forget gate(f_t), Input gate(i_t), Output gate(o_t)



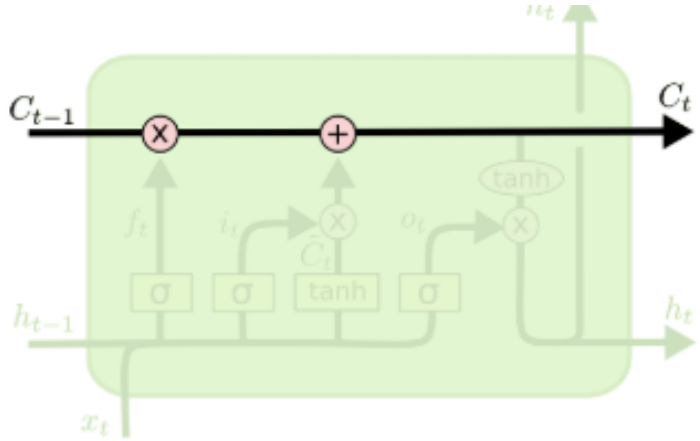
Cell state (C_t)

Hidden state (h_t)

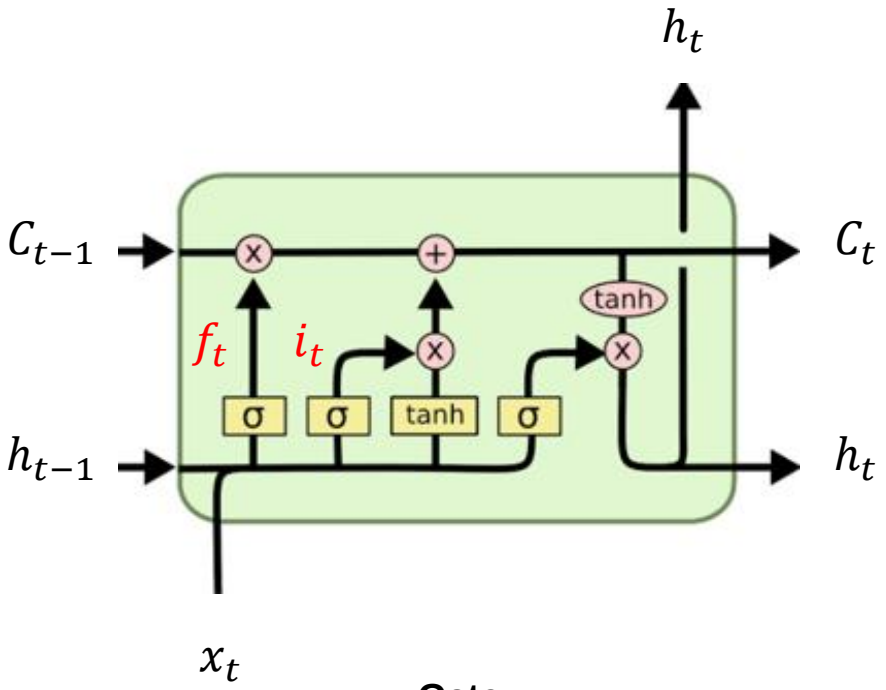
LSTM

LSTM

- **Cell state (C_t)**
  - LSTM의 핵심 구조로써, **장기적인 정보 (Long term)들을 유지**
  - 두가지 gate (Forget gate(f_t), Input gate(i_t))를 통해 cell state 업데이트



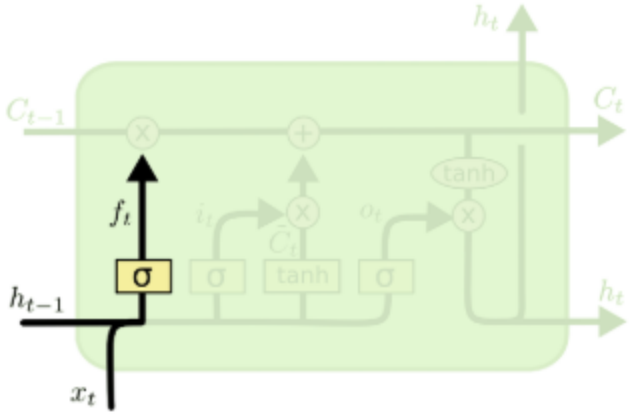
Cell state



Gate

LSTM

- **Cell state (C_t)**
  - Forget gate(f_t): 불필요한 과거 정보를 잊기 위한 gate



Forget gate

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

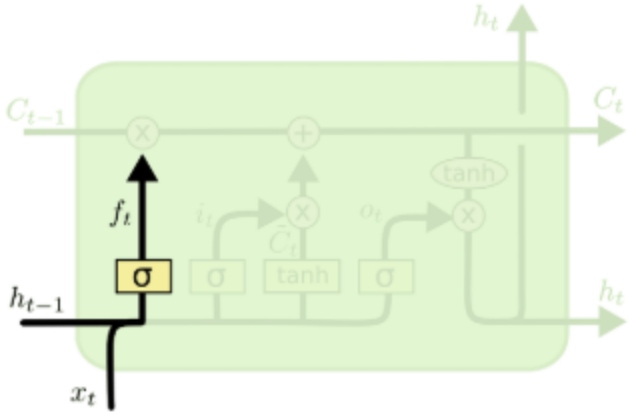
Sigmoid (0 ~ 1 사이 가중치)

다 잊는 경우

f_t	0	0	0	0	0	0	0
C_{t-1}	0.2	0.1	0.3	0.5	0.2	0.2	0.3
$f_t \otimes C_{t-1}$	0	0	0	0	0	0	0

LSTM

- **Cell state (C_t)**
  - Forget gate(f_t): 불필요한 과거 정보를 잊기 위한 gate



Forget gate

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

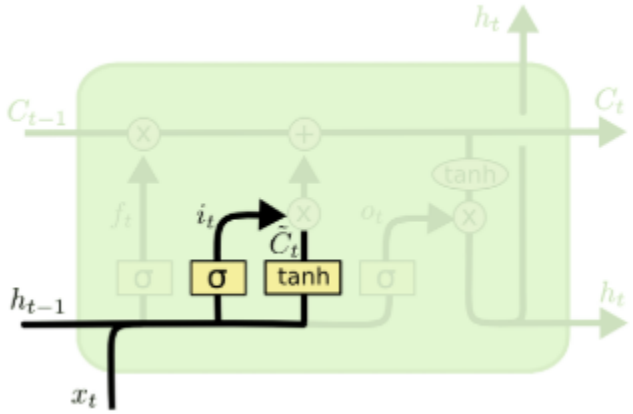
Sigmoid (0 ~ 1 사이 가중치)

모두 기억하는 경우

f_t	1	1	1	1	1	1	1
C_{t-1}	0.2	0.1	0.3	0.5	0.2	0.2	0.3
$f_t \otimes C_{t-1}$	0.2	0.1	0.3	0.5	0.2	0.2	0.3

LSTM

- **Cell state (C_t)**
  - Forget gate(f_t): 불필요한 과거 정보를 잊기 위한 gate
  - Input gate(i_t): 현재 정보를 기억하기 위한 gate



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

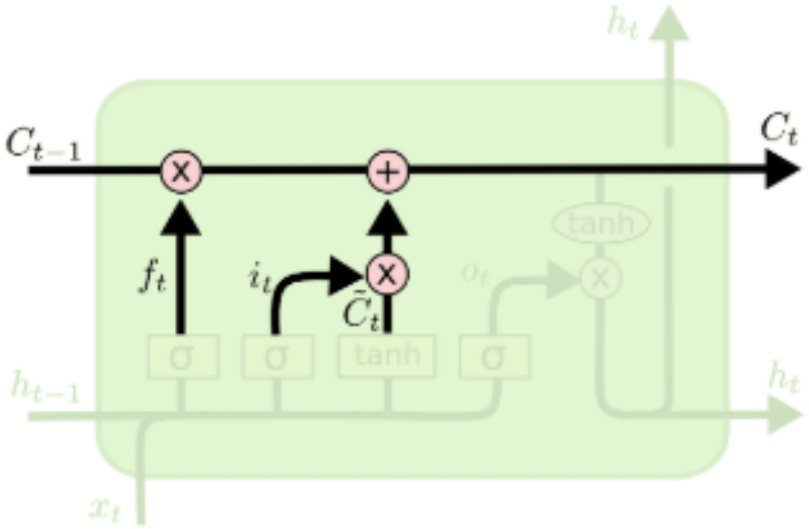
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

i_t	0.1	0	0.8	0.2	0.8	0.7	1
$\tilde{C}_t$	0.1	0.3	0.6	0.2	0.9	0.1	0.4
$i_t \otimes \tilde{C}_t$	0.01	0	0.48	0.04	0.72	0.07	0.4

Input gate

LSTM

- **Cell state (C_t)**
  - Forget gate(f_t): 불필요한 과거 정보를 잊기 위한 gate
  - Input gate(i_t): 현재 정보를 기억하기 위한 gate
  - Cell state (C_t) = 불필요한 정보를 제거한 이전 시점의 cell state (C_{t-1}) + 현재 시점의 cell state ($\tilde{C}_t$)

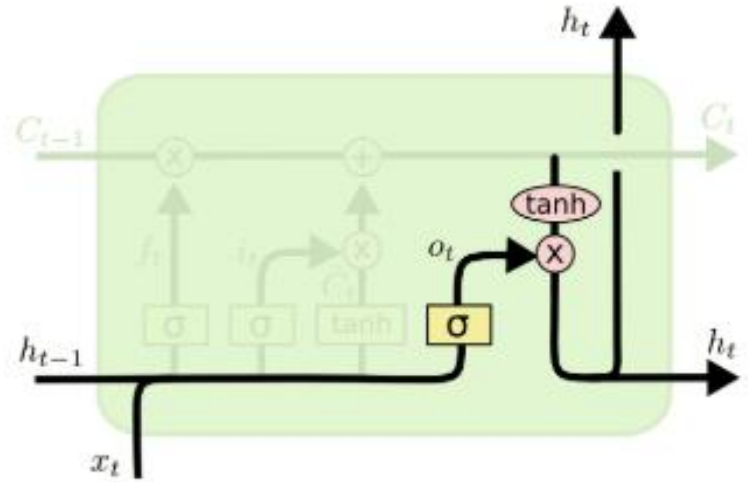


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Cell state 업데이트

LSTM

- **Hidden state (h_t)**
  - 단기적인 정보 (Short term)을 유지
  - Output gate(o_t): Hidden state에 cell state를 얼마나 반영할 것인지에 대한 가중치



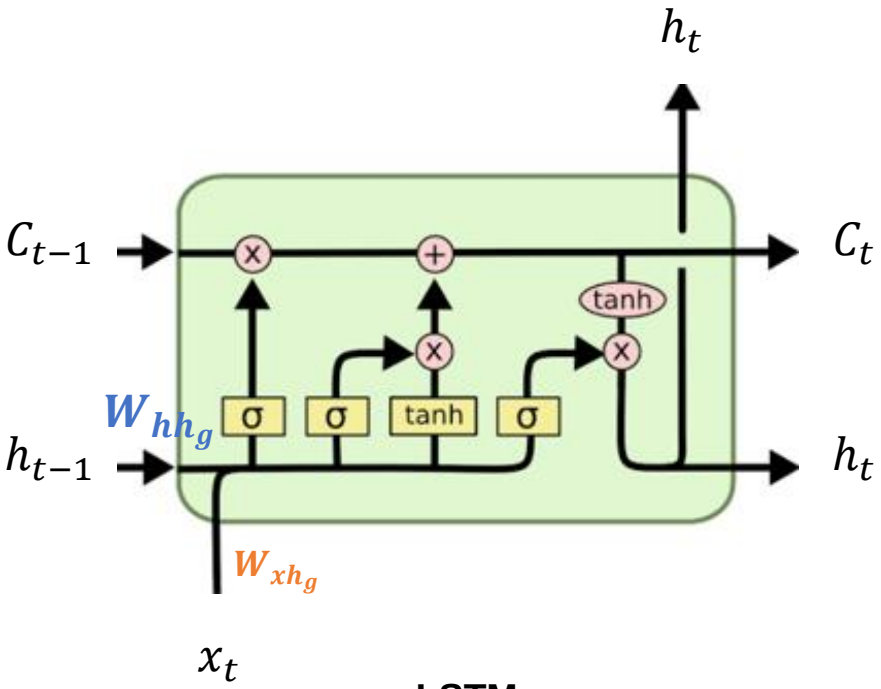
$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

Hidden state 업데이트

LSTM

- **Review**
  - Cell state (C_t): 현 시점에 대한 장기적인 정보 (Long term)들을 유지
  - Hidden state (h_t): 현 시점에 대한 단기적인 정보 (Short term)을 유지



LSTM

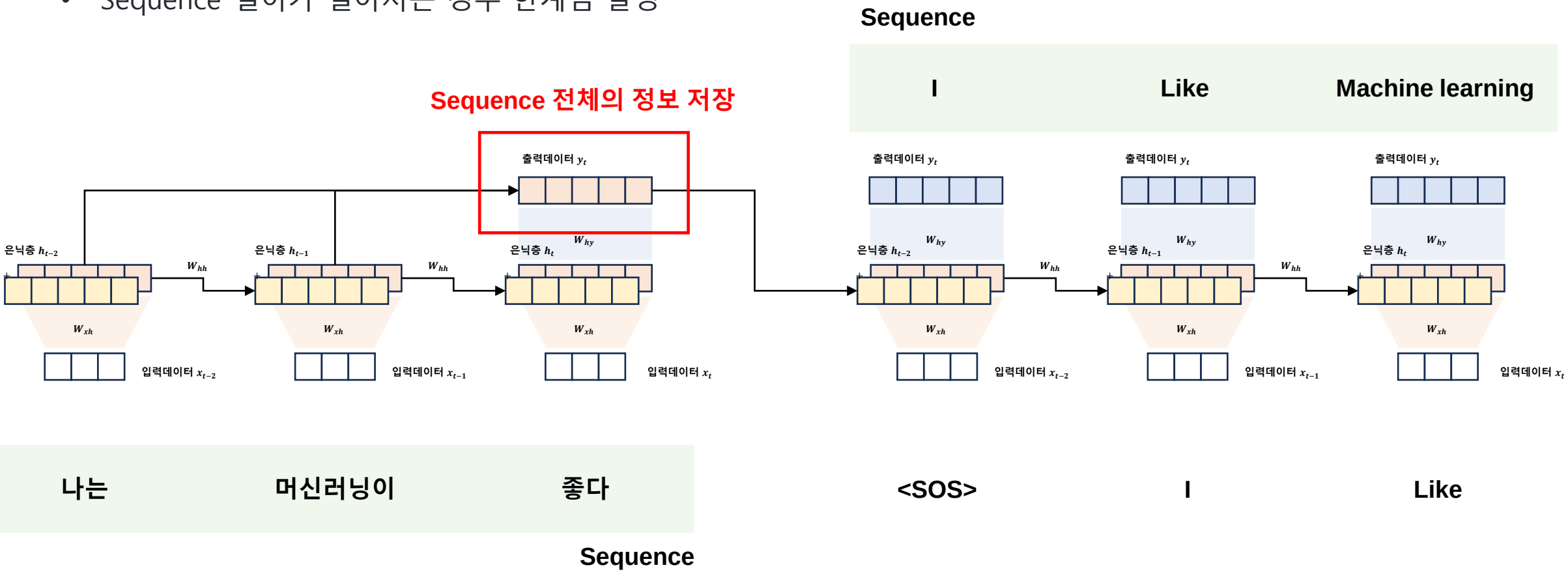
$$f_t = \sigma(W_{xhf}x_t + W_{hhf}h_{t-1} + bias)$$

$$i_t = \sigma(W_{xhi}x_t + W_{hhi}h_{t-1} + bias)$$

$$o_t = \sigma(W_{xho}x_t + W_{hho}h_{t-1} + bias)$$

LSTM의 한계점 및 Transformer

- **장기의존성 문제**
  - LSTM을 이용하여 문제를 완화하였으나 여전히 장기의존성 문제가 존재함 → Seq2Seq 방법으로 완화함
  - Sequence 길이가 길어지는 경우 한계점 발생



LSTM의 한계점 및 Transformer

- **장기의존성 문제**
  - LSTM을 이용하여 문제를 완화하였으나 여전히 장기의존성 문제가 존재함
  - Sequence 길이가 길어지는 경우 한계점 발생

- **병렬처리 문제**
  - RNN 및 LSTM은 순차적으로 입력, 출력하는 구조이기 때문에 병렬처리가 어려움

- **Transformer**
  - Self attention 기법을 활용한 transformer가 위의 문제점을 해결함
  - 이후 많은 분야에서 transformer를 이용한 연구 진행 중

Word2Vec

각각의 방법론은 단어를 벡터로 변환하는 메커니즘이 조금씩 다릅니다.

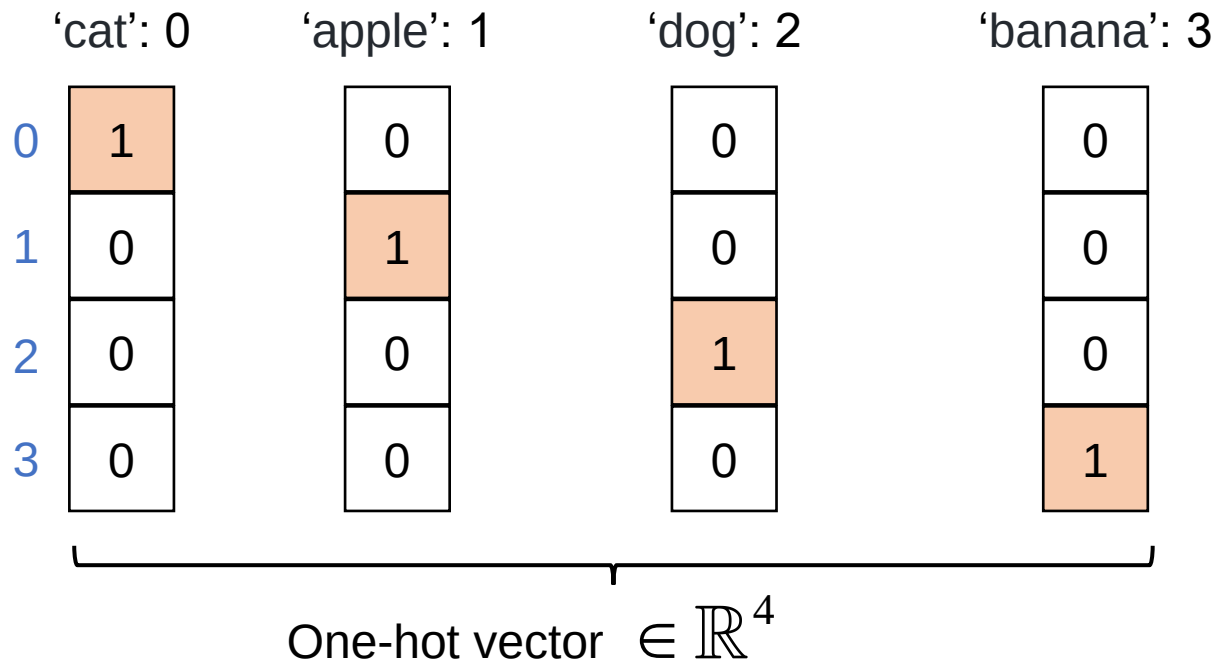
Word2Vec

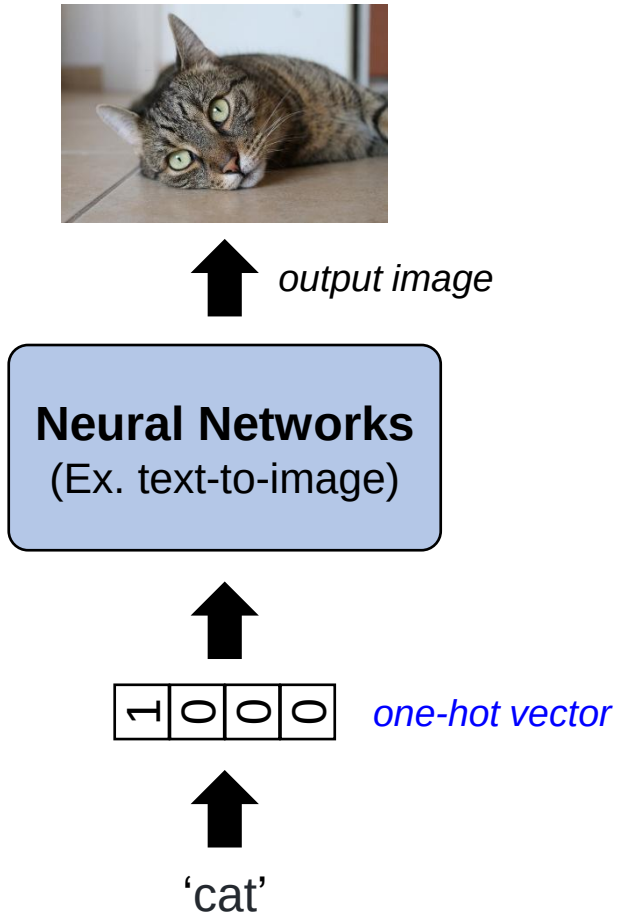
GloVe

FastText

Background – One-hot Encoding

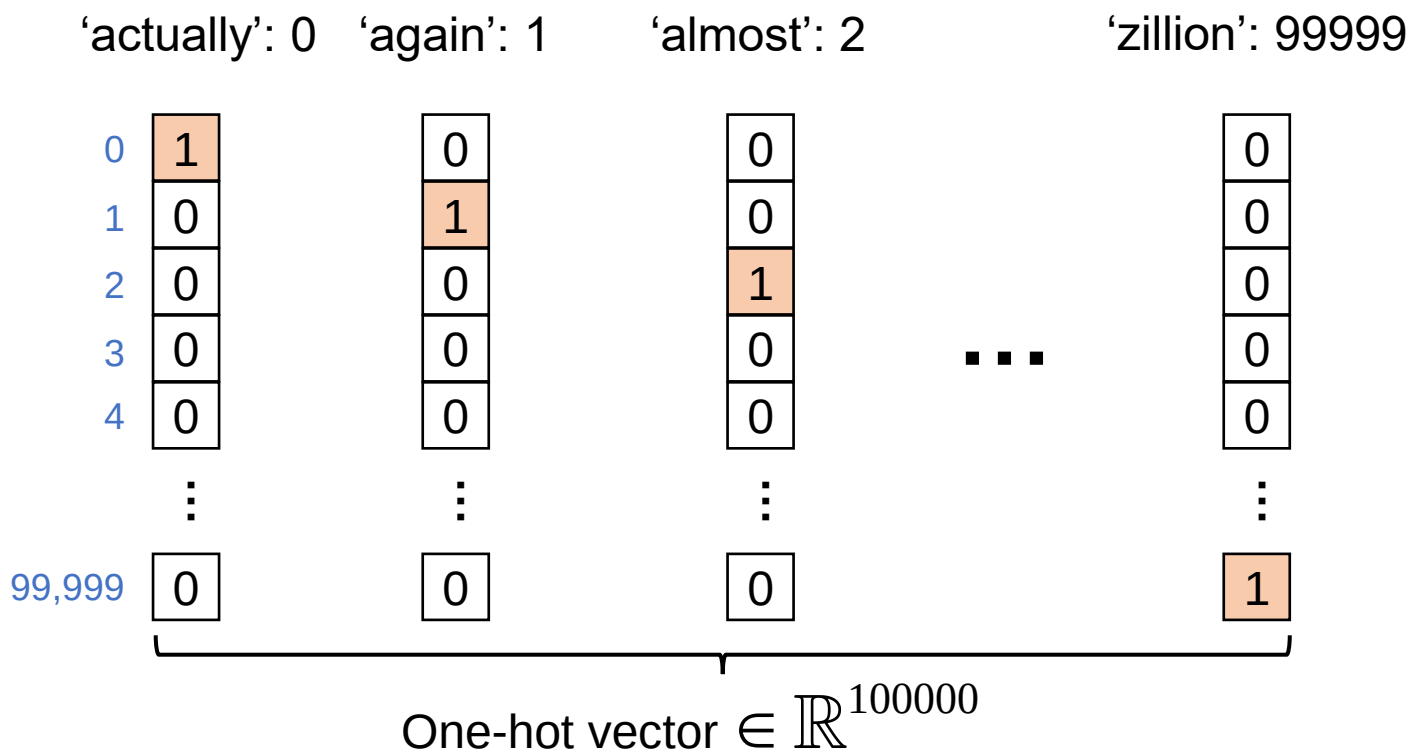
- **전통적인 언어 데이터 표현 방법: One-hot Encoding**
  - 각 단어를 고유한 인덱스로 변환한 후 one-hot vector로 표현
  - Ex. 4개 단어에 대한 one-hot encoding
    - cat, apple, dog, banana





Background – One-hot Encoding

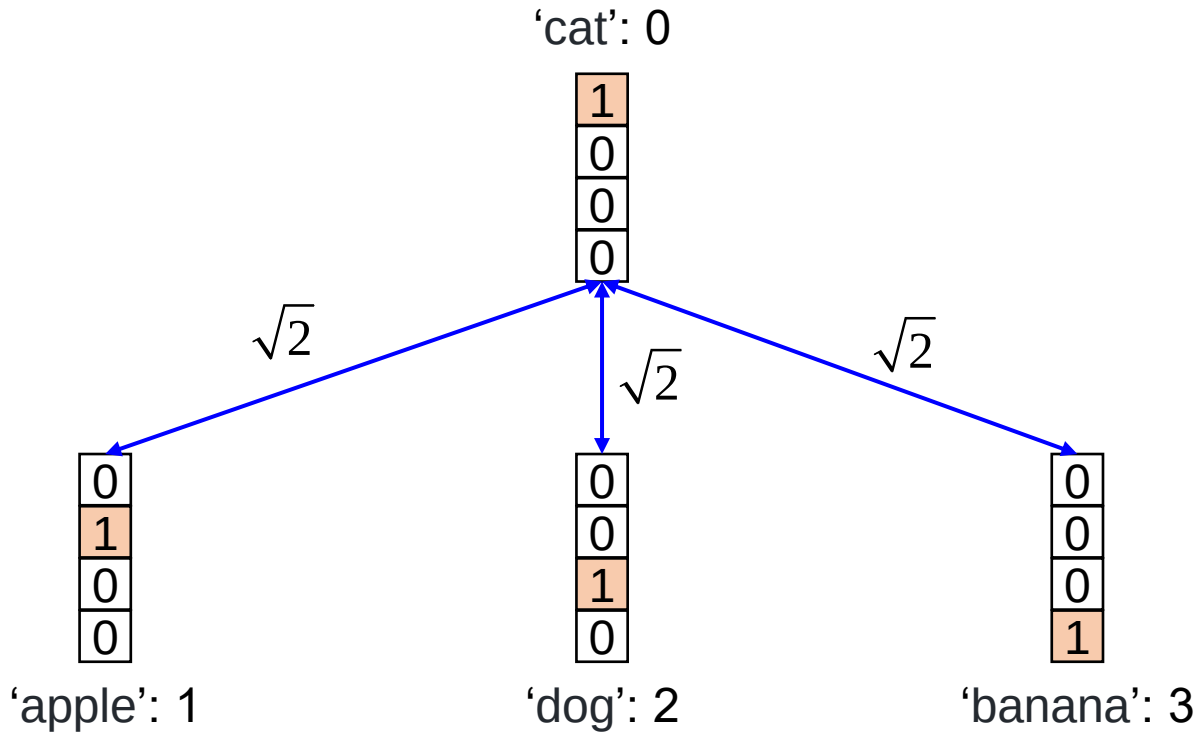
- **One-hot Encoding 기반 단어 표현의 한계 (1)**
  - 단어의 개수가 많아질수록 one-hot vector의 크기가 커짐 → 메모리 요구량 및 계산 효율성 문제
  - Ex. 100,000개 단어에 대한 one-hot encoding



Background – One-hot Encoding

- **One-hot Encoding 기반 단어 표현의 한계 (2)**
  - 단어 간의 관계 또는 유사도 식별 불가 → 단어가 가지는 특징을 설명하지 못함
  - Ex. 4개 단어에 대한 one-hot vector 유사도 비교

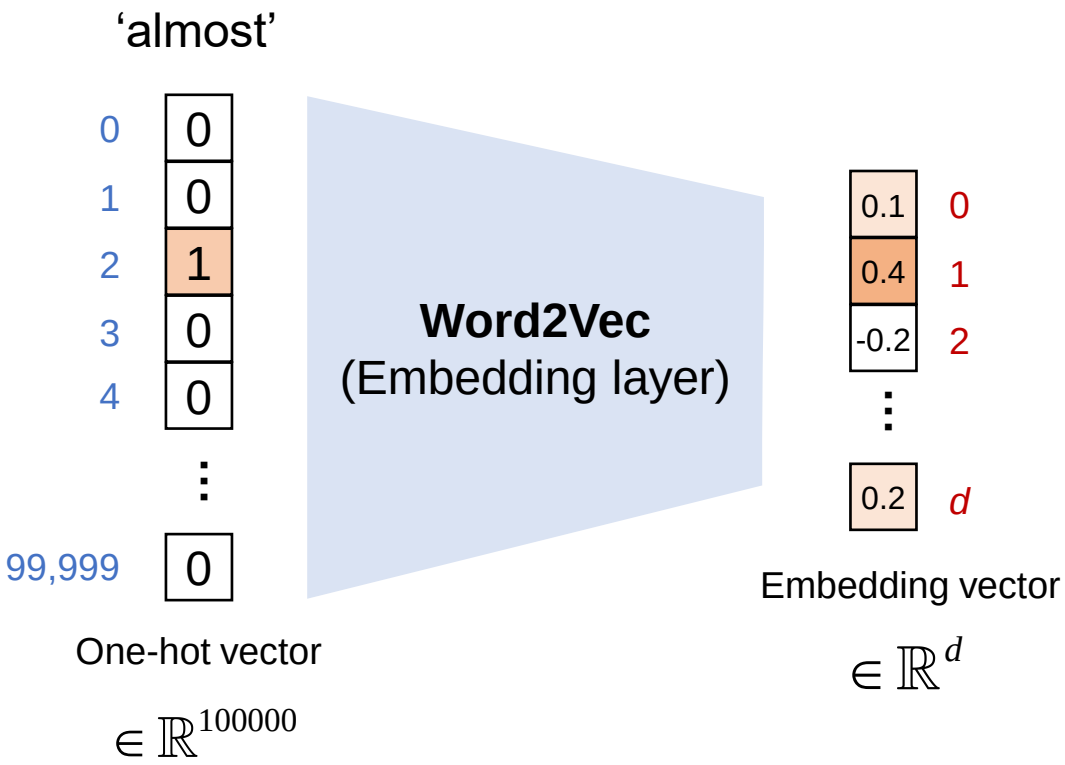
$$\text{L2\_distance}(A, B) = \|A - B\|_2 = \sqrt{\sum_{i=0}^{N-1} (A_i - B_i)^2}$$

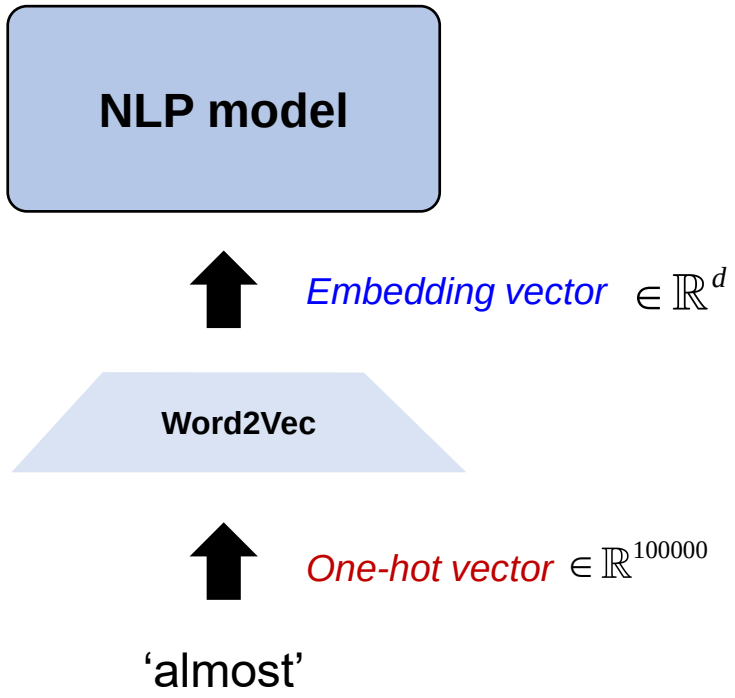


One-hot vector 간 유사도는 항상 동일한 값을 가짐

Word2Vec

- **One-hot Encoding 기반 단어 표현의 한계 극복: Word2Vec**
  - 각 단어를 낮은 차원의 실수 벡터 (embedding vector)로 표현 → 계산 효율성 문제 해결
  - Ex. 100,000개 단어에 대한 Word2Vec

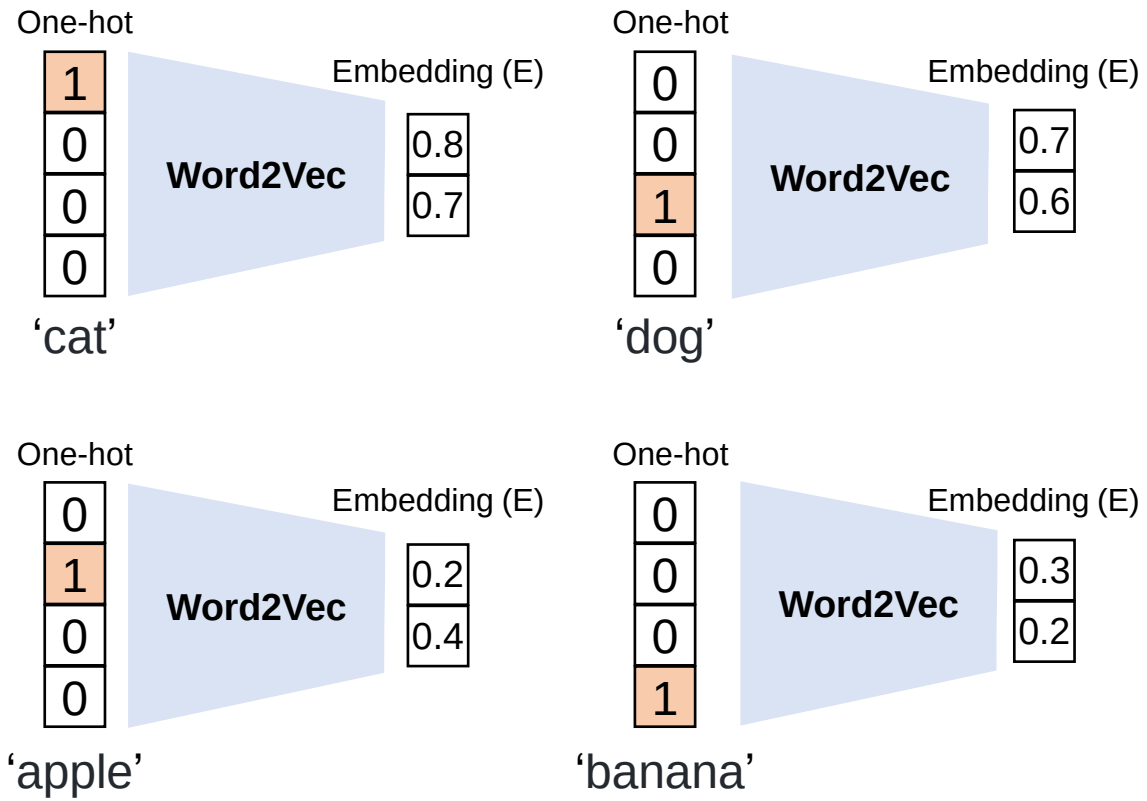




❖ d: Embedding vector의 dimension (Hyper-parameter)

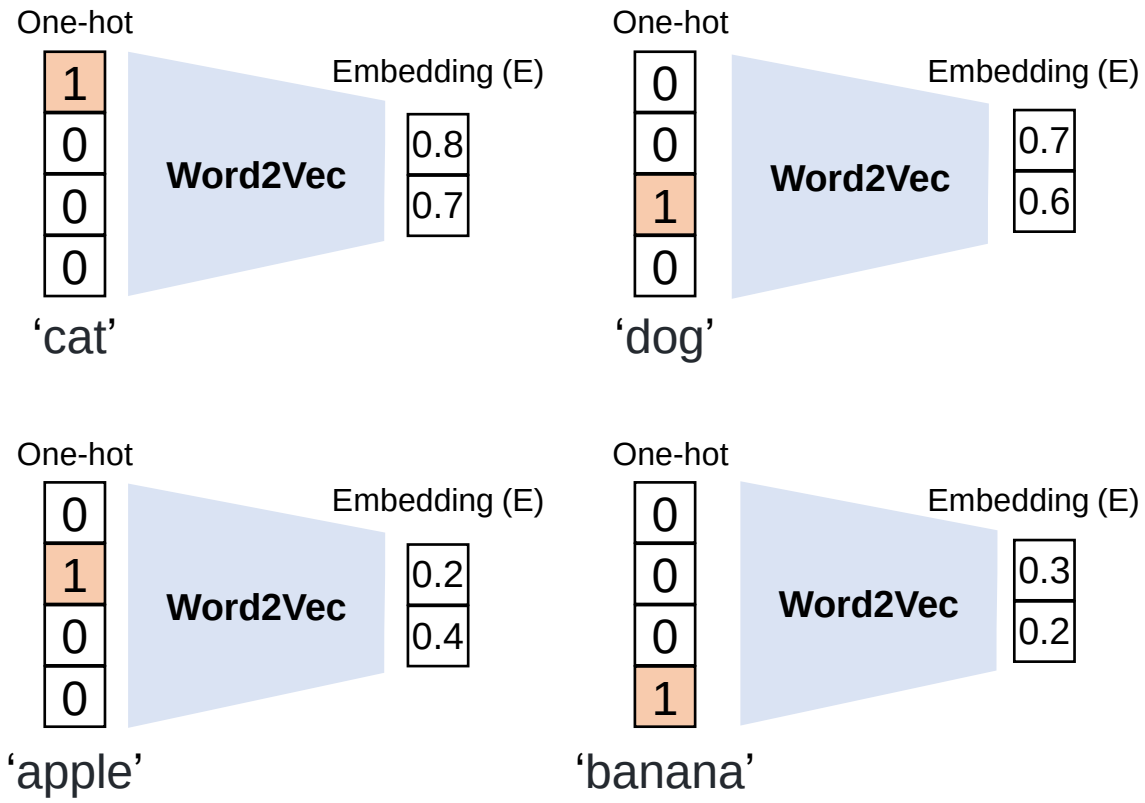
Word2Vec

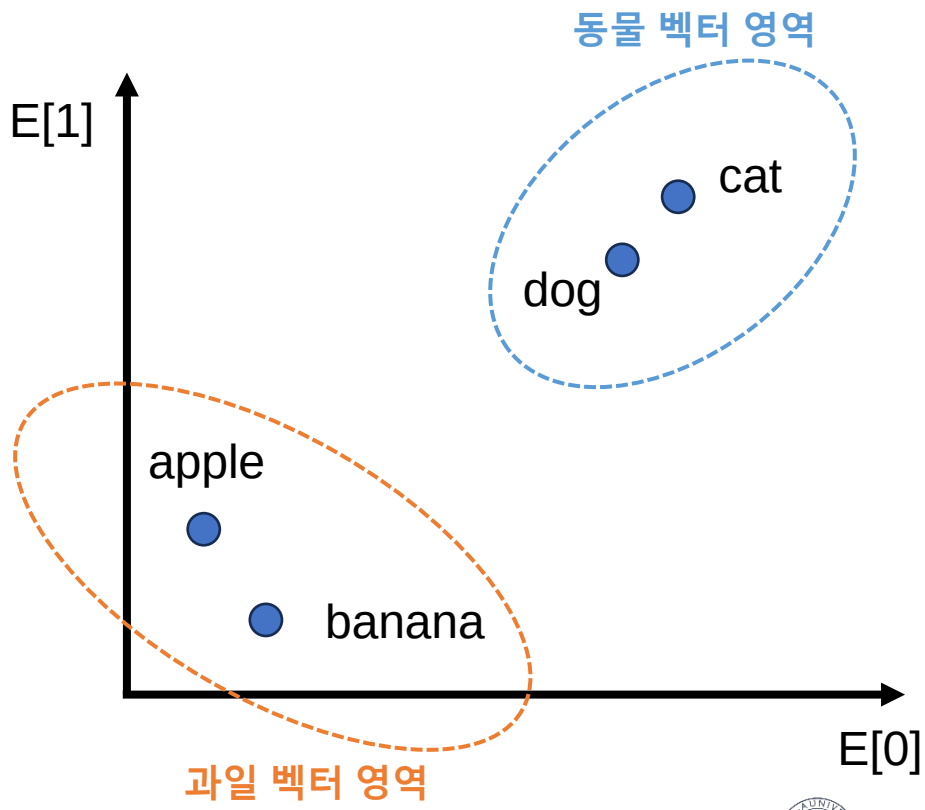
- **One-hot Encoding 기반 단어 표현의 한계 극복: Word2Vec**
  - 각 단어를 낮은 차원의 실수 벡터 (embedding vector)로 표현



Word2Vec

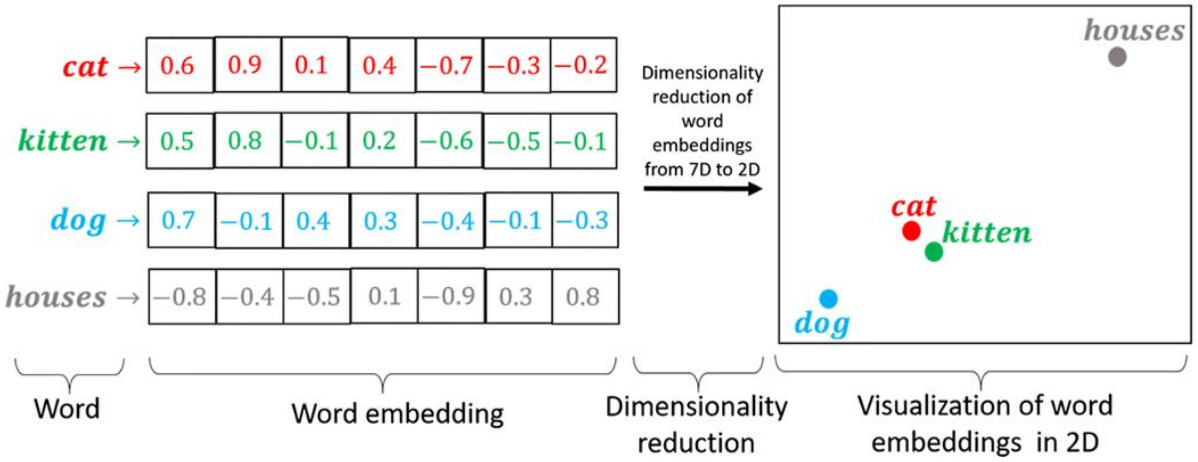
- **One-hot Encoding 기반 단어 표현의 한계 극복: Word2Vec**
  - 각 단어를 낮은 차원의 실수 벡터 (embedding vector)로 표현
  - Word2Vec가 잘 학습된 경우 벡터간 유사도 비교를 통해 단어 간 관계 도출 가능



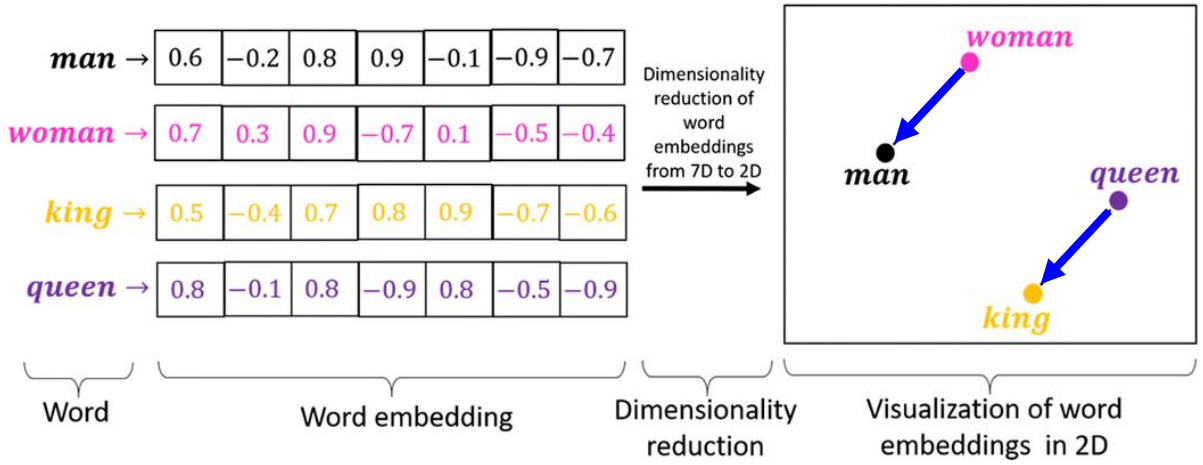


Word2Vec

- **One-hot Encoding 기반 단어 표현의 한계 극복: Word2Vec**
  - 각 단어를 낮은 차원의 실수 벡터 (embedding vector)로 표현
  - Word2Vec가 잘 학습된 경우 벡터간 유사도 비교를 통해 단어 간 관계 도출 가능
  - 단어 간 유사도를 거리, 관계를 벡터로 표현 가능



❖ **유사한 단어간 거리 차이 예시**



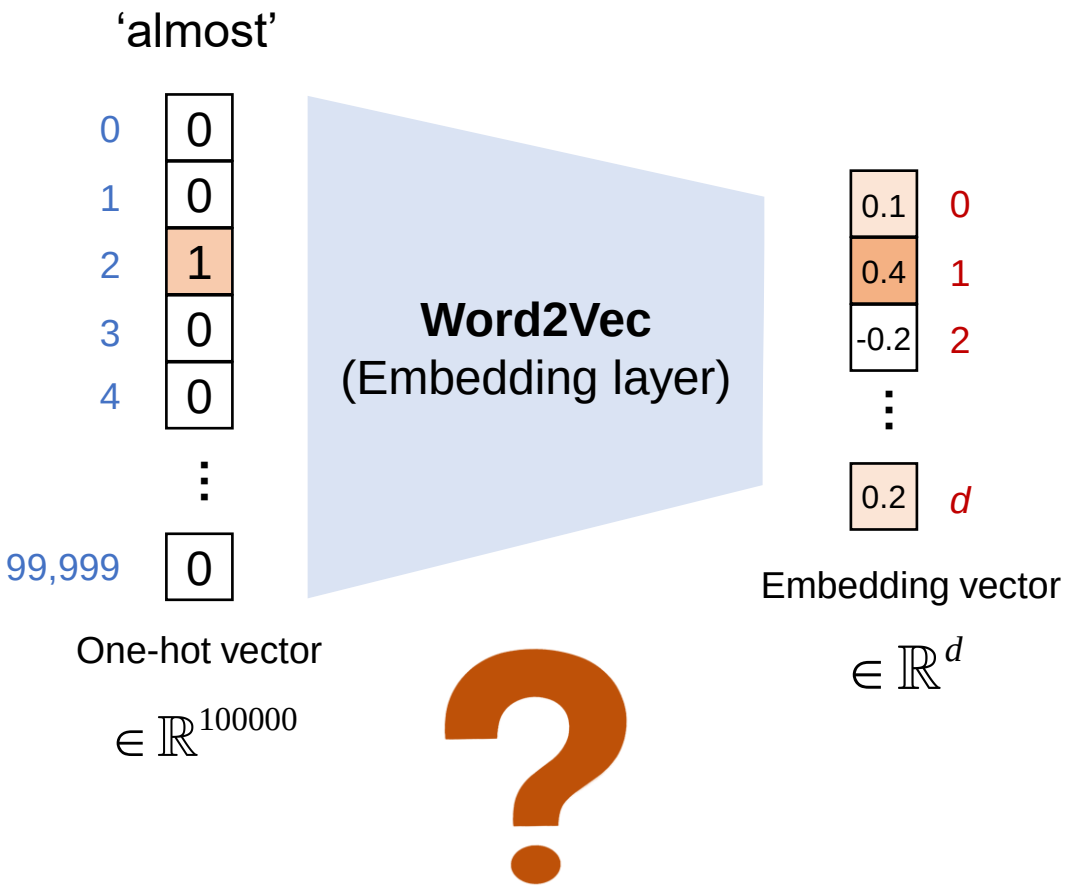
❖ **단어 간 관계를 벡터로 표현한 예시**

✓ woman ➔ man 벡터와, queen ➔ king 벡터는 유사한 형태를 가짐

Ref.: https://medium.com/@hari4om/word-embedding-d816f643140

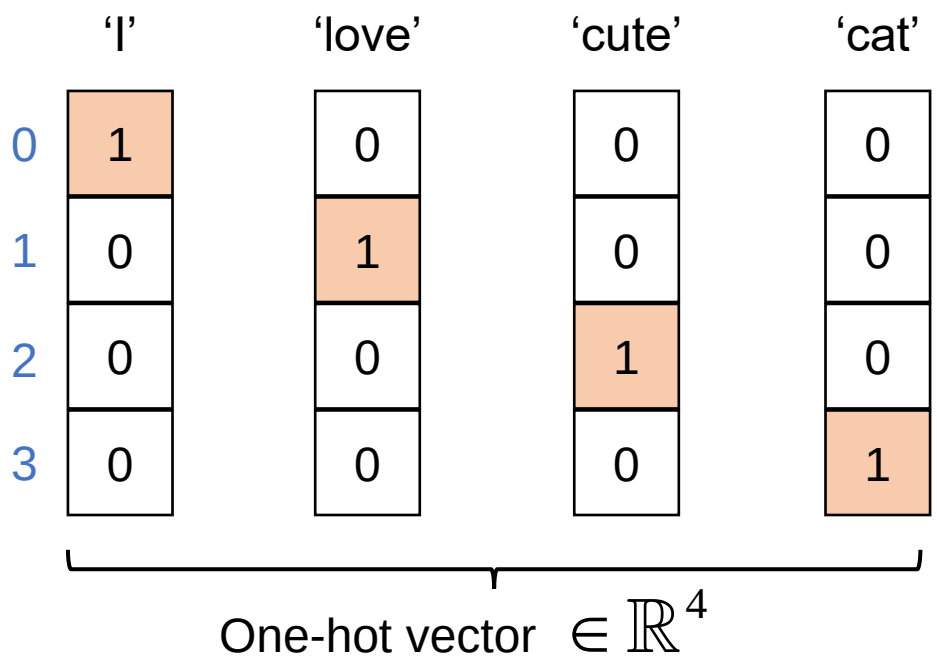
Word2Vec

- **One-hot Encoding 기반 단어 표현의 한계 극복: Word2Vec**

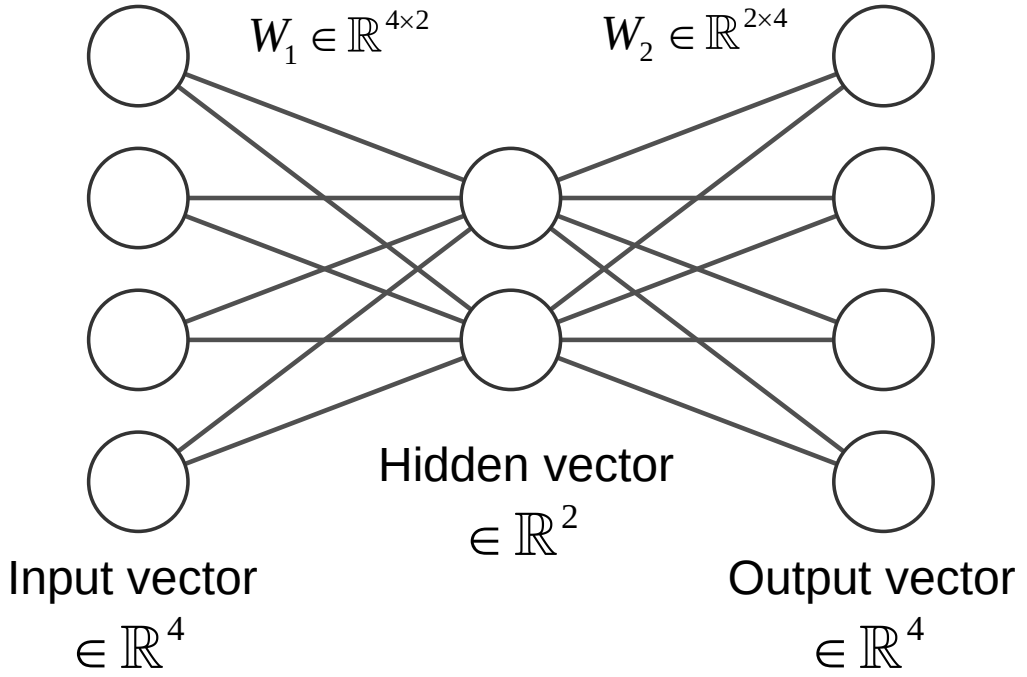


Word2Vec

- **Word2Vec 모델 학습 – 단어 간 독립 학습**
  - Ex. 4개 단어에 대한 Word2Vec 모델



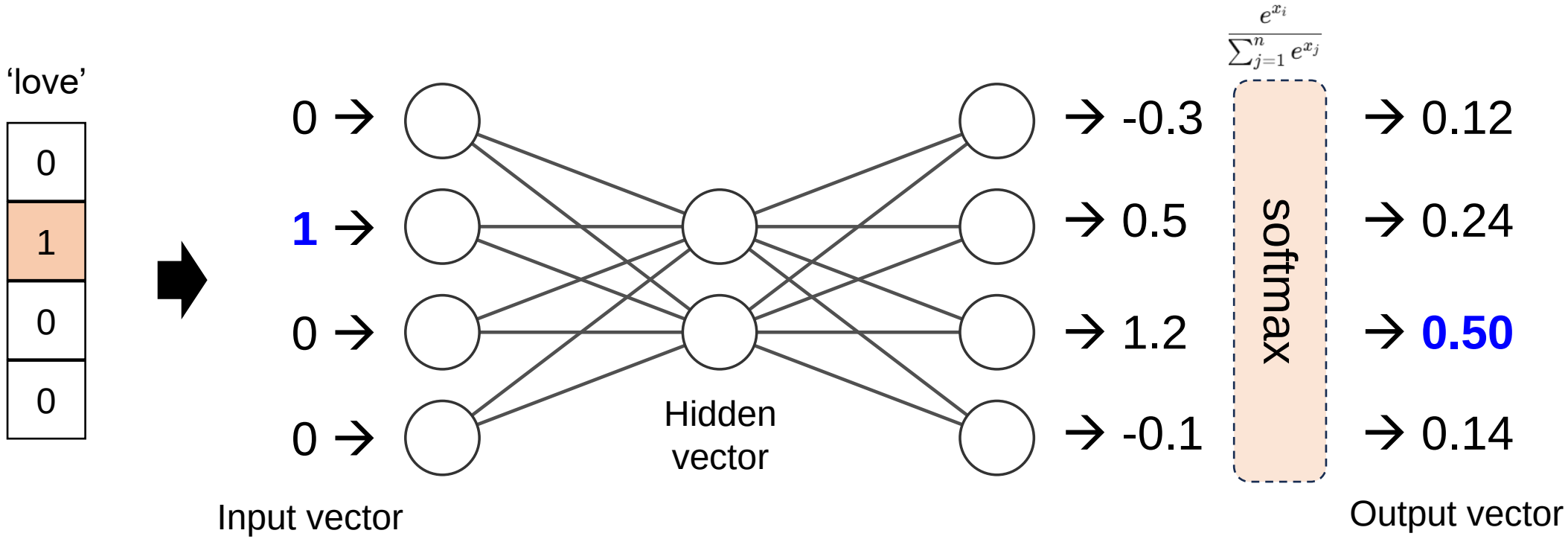
Word2Vec 모델 예시



❖ Hidden node의 차원 (d)는 hyperparameter

Word2Vec

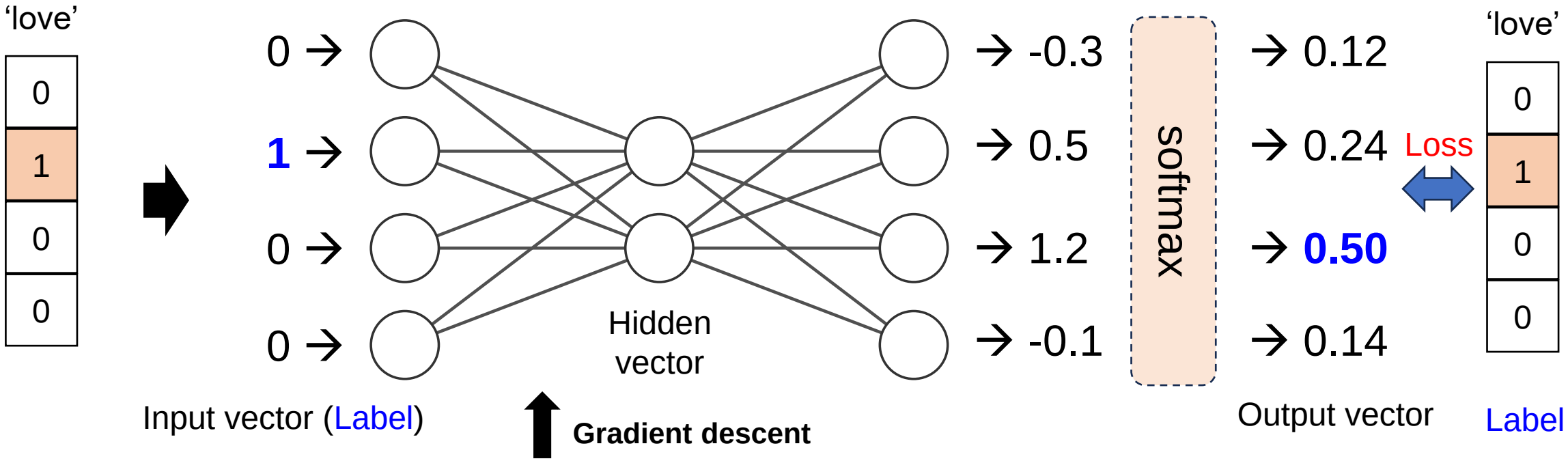
- **Word2Vec 모델 학습 – 단어 간 독립 학습**
  - Ex. 4개 단어에 대한 Word2Vec 모델: 충분히 학습되지 않은 경우



Word2Vec

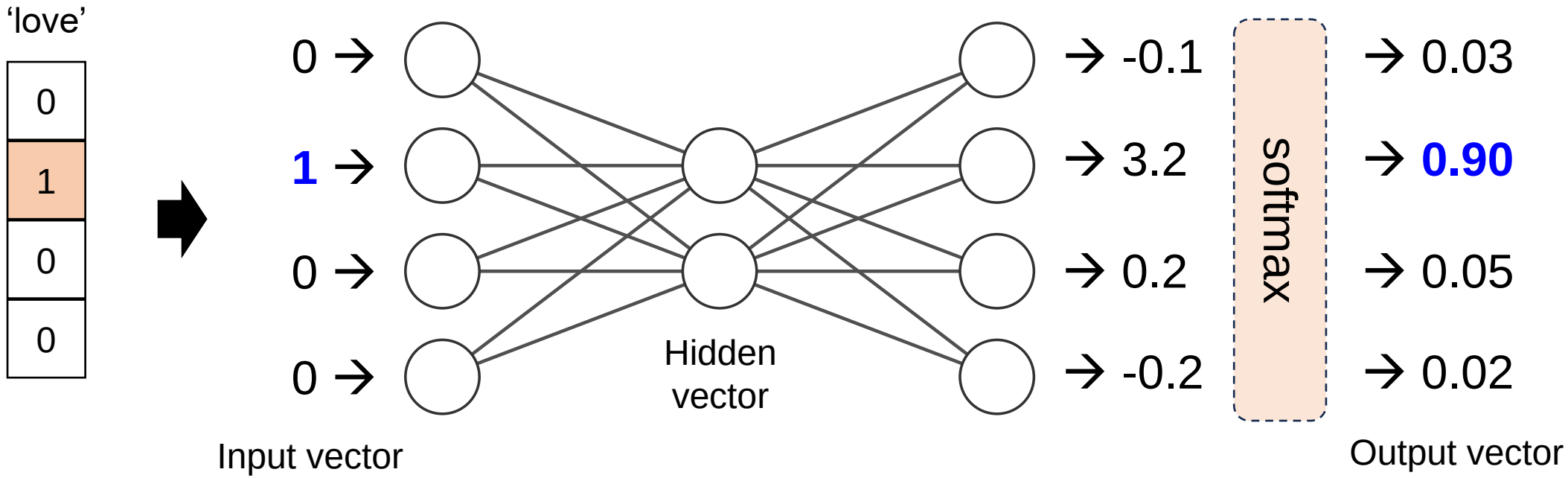
- **Word2Vec 모델 학습 – 단어 간 독립 학습**
  - Ex. 4개 단어에 대한 Word2Vec 모델: 충분히 학습되지 않은 경우

$$CE(Y,\hat{Y}) = -\sum_{i=0}^{N-1} Y_i \log(\hat{Y}_i)$$



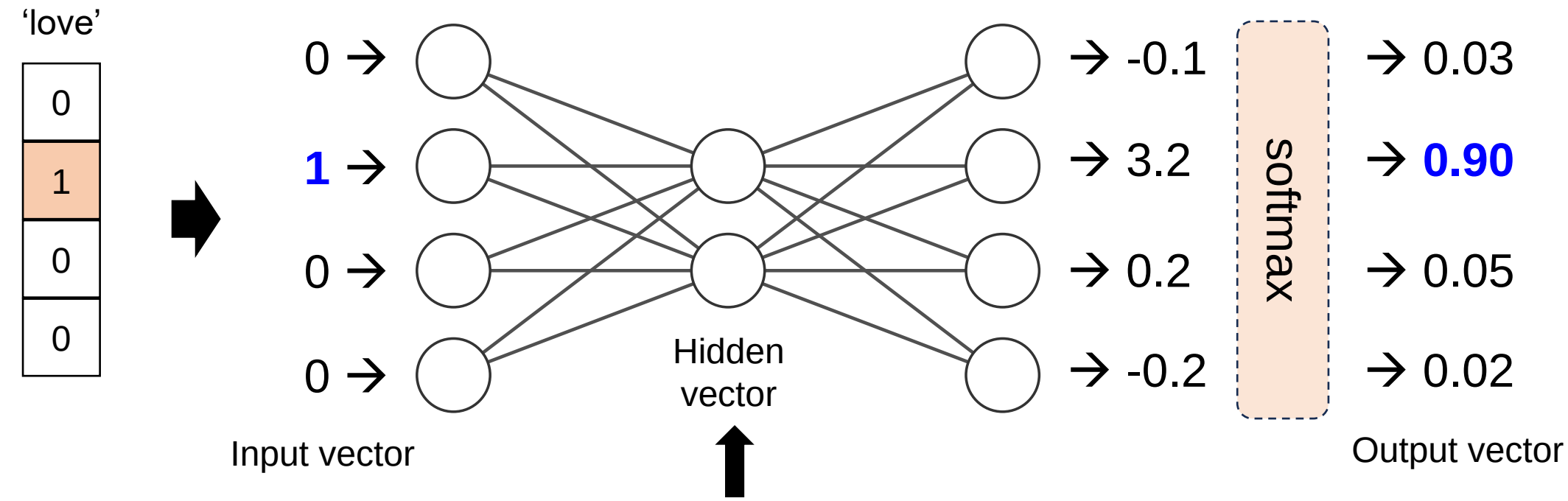
Word2Vec

- **Word2Vec 모델 학습 – 단어 간 독립 학습**
  - Ex. 4개 단어에 대한 Word2Vec 모델: 충분히 학습된 경우



Word2Vec

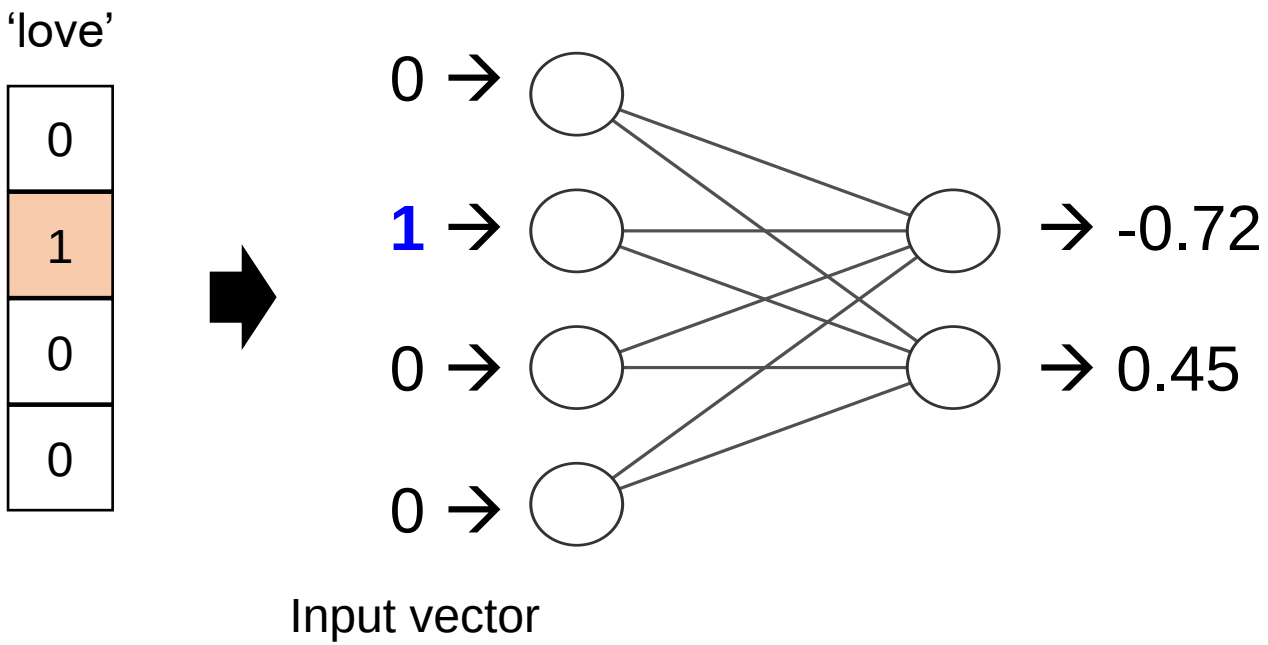
- **Word2Vec 모델 학습 – 단어 간 독립 학습**
  - Ex. 4개 단어에 대한 Word2Vec 모델: 충분히 학습된 경우



Hidden vector가 단어 'love'에 대해 충분히 잠재적인 정보를 가지고 있다고 판단

Word2Vec

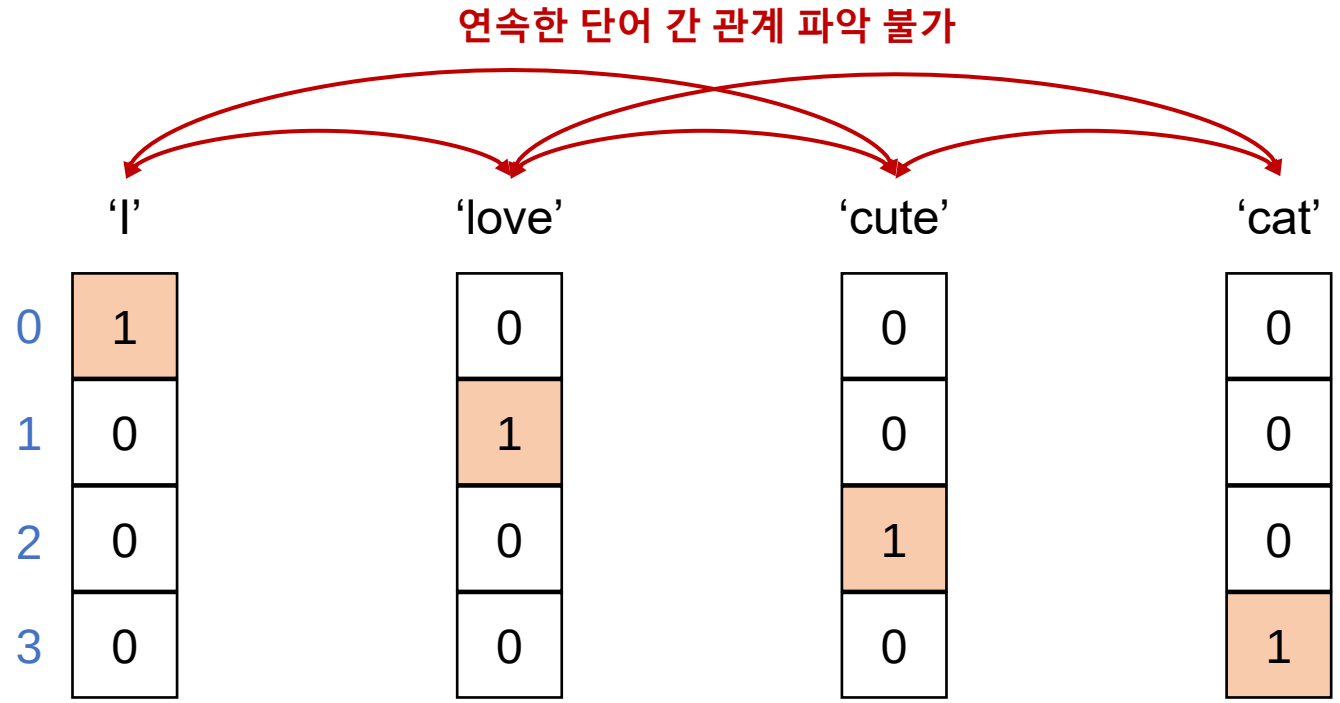
- **Word2Vec 모델 학습 – 단어 간 독립 학습**
  - Ex. 4개 단어에 대한 Word2Vec 모델: 충분히 학습된 경우



(중요: Embedding vector는 훈련이 끝난 Weight에 대한 각 행의 Weight로 표현됨)

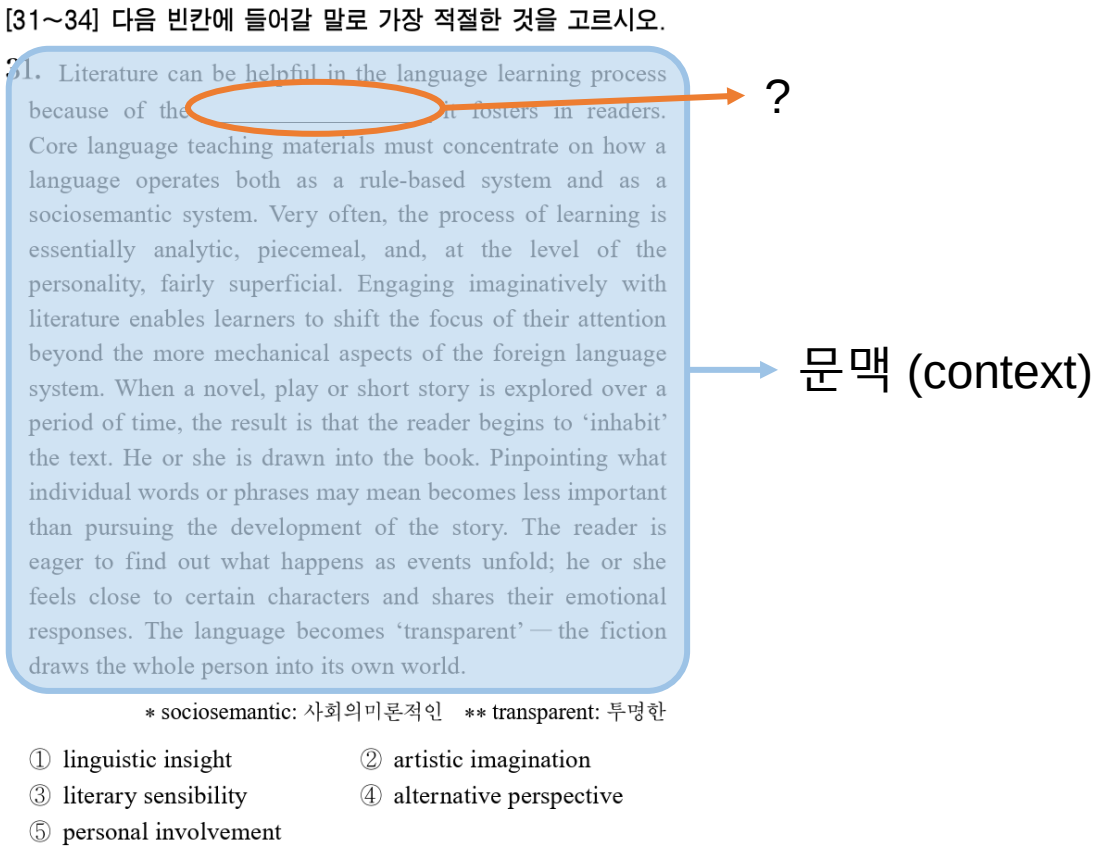
Word2Vec

- **Word2Vec 모델 학습 – 단어 간 독립 학습**
  - 단어별로 독립적인 변환 방법을 학습 → 여러 단어 간 관계를 파악할 수 없음
    - ✓ 해결방법 (1): Continuous Bag-of-words (CBOW)
    - ✓ 해결방법 (2): Skip-Gram



Word2Vec

- **Word2Vec 모델 학습 – Continuous Bag-of-words (CBOW)**
  - 빈칸 추론 문제에서 전체적인 문맥 (context)를 통해 빈칸에 들어갈 정답을 찾도록 학습



[31~34] 다음 빈칸에 들어갈 말로 가장 적절한 것을 고르시오.

31. Literature can be helpful in the language learning process because of the _____ it fosters in readers. Core language teaching materials must concentrate on how a language operates both as a rule-based system and as a sociosemantic system. Very often, the process of learning is essentially analytic, piecemeal, and, at the level of the personality, fairly superficial. Engaging imaginatively with literature enables learners to shift the focus of their attention beyond the more mechanical aspects of the foreign language system. When a novel, play or short story is explored over a period of time, the result is that the reader begins to 'inhabit' the text. He or she is drawn into the book. Pinpointing what individual words or phrases may mean becomes less important than pursuing the development of the story. The reader is eager to find out what happens as events unfold; he or she feels close to certain characters and shares their emotional responses. The language becomes 'transparent' — the fiction draws the whole person into its own world.

* sociosemantic: 사회의미론적인 ** transparent: 투명한

① linguistic insight ② artistic imagination
③ literary sensibility ④ alternative perspective
⑤ personal involvement

I ??? cat

We ??? movie

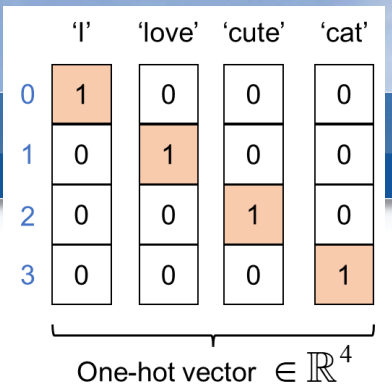
You ??? picture

Cat ??? dog

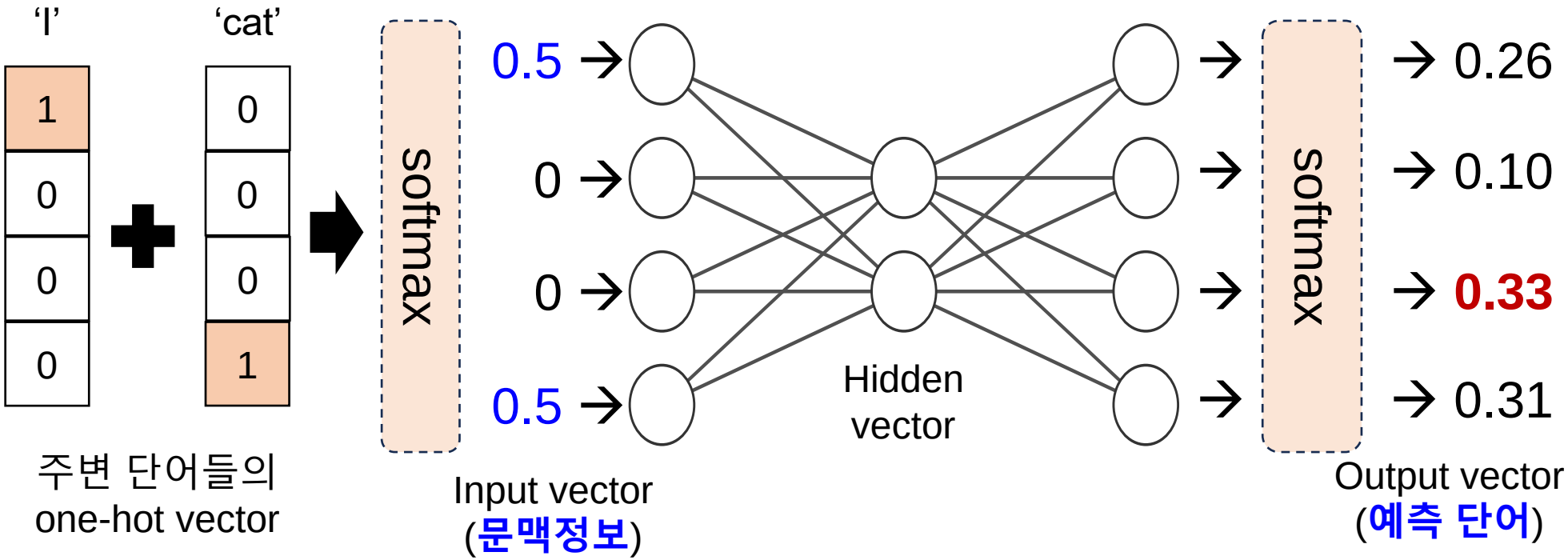
CBOW는 전후 문맥을 고려해 중간에 들어 갈 단어를 학습

Ref.: https://www.suneung.re.kr/boardCnts/list.do?boardID=1500234&m=0403&s=suneung#;

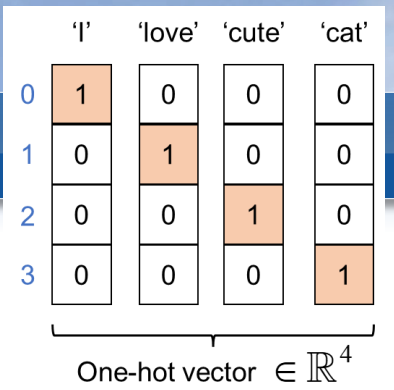
Word2Vec



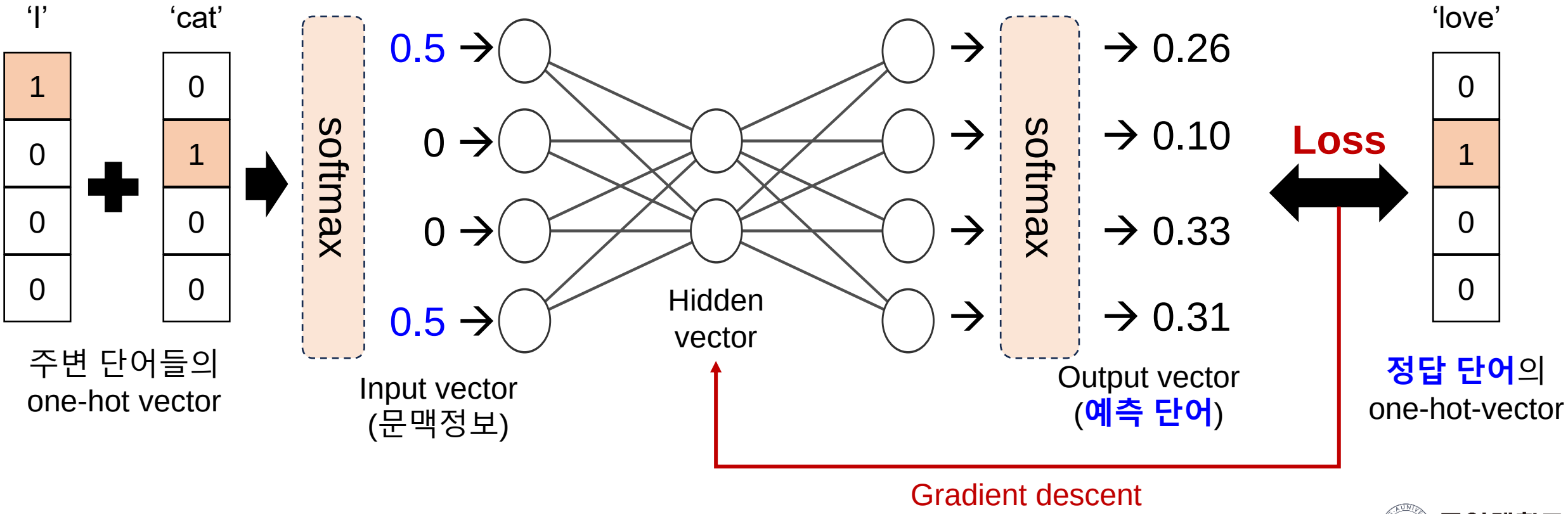
- **Word2Vec 모델 학습 – Continuous Bag-of-words (CBOW)**
  - Ex. I love cat 문장 중 'love'를 예측 하도록 학습



Word2Vec

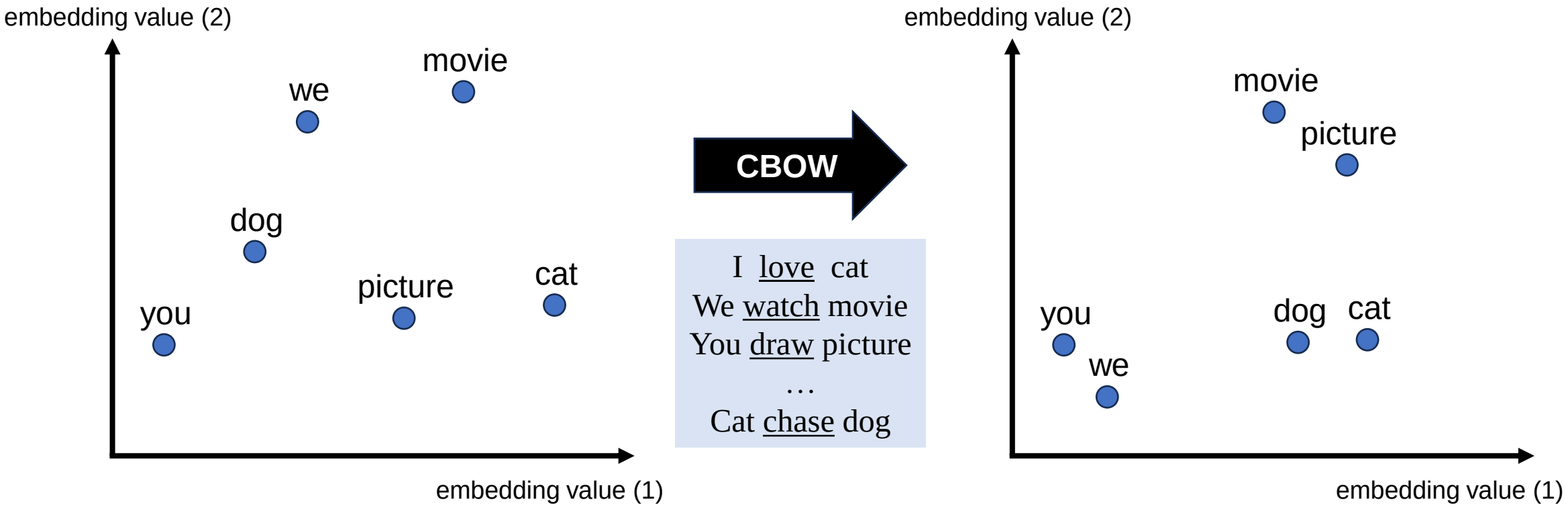


- **Word2Vec 모델 학습 – Continuous Bag-of-words (CBOW)**
  - Ex. I love cat 문장 중 'love'를 예측 하도록 학습



Word2Vec

- **Word2Vec 모델 학습 – Continuous Bag-of-words (CBOW)**
  - 단어들이 문맥 안에서 관계성을 스스로 학습
  - 대규모 문장 데이터만 있다면 Word2Vec 학습 가능

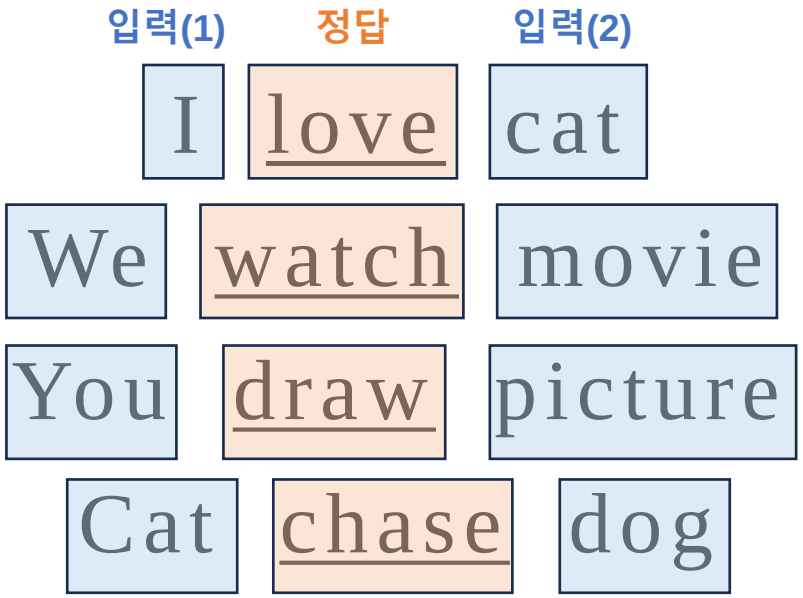


단어 간 독립 학습

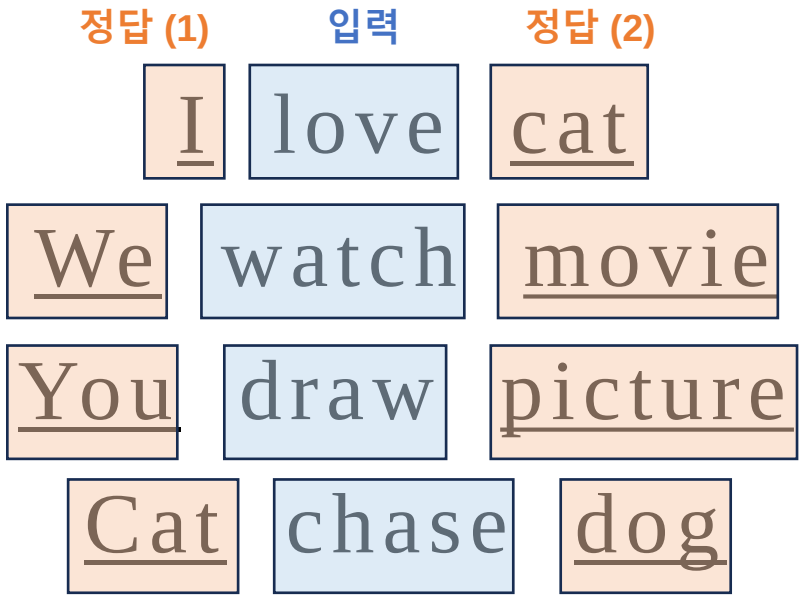
CBOW 기반 학습

Word2Vec

- **Word2Vec 모델 학습 – Skip-Gram**
  - CBOW와 반대 개념의 의미를 학습
    - ✓ CBOW: 여러 개 단어를 주고 하나의 단어를 출력
    - ✓ Skip-Gram: 하나의 단어를 주고 그 단어 주변에 나타날 수 있는 여러 단어 출력



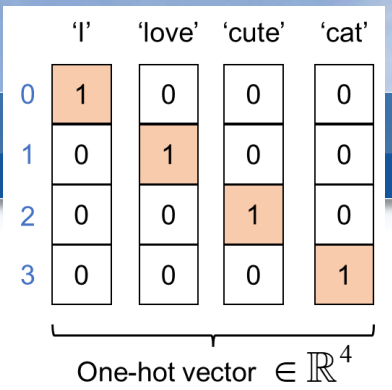
CBOW 학습 데이터 구성 예시

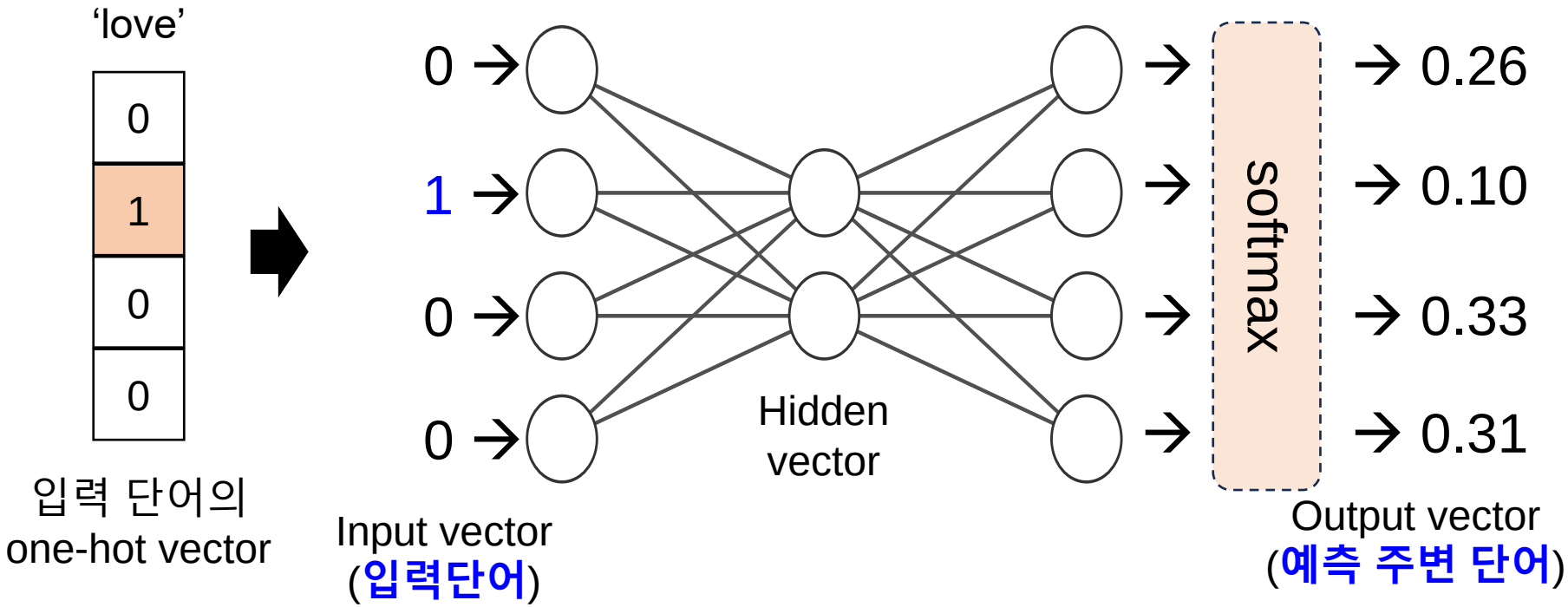


Skip-Gram 학습 데이터 구성 예시

Word2Vec

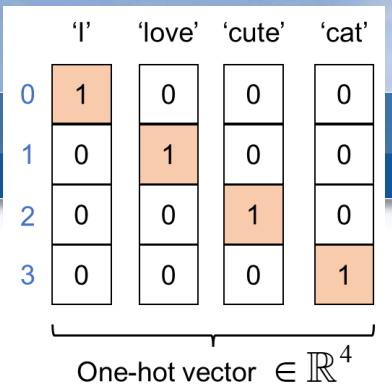
- **Word2Vec 모델 학습 – Skip-Gram**
  - Ex. I love cat 문장 중 'love' 주변에 나올 수 있는 단어를 예측 하도록 학습

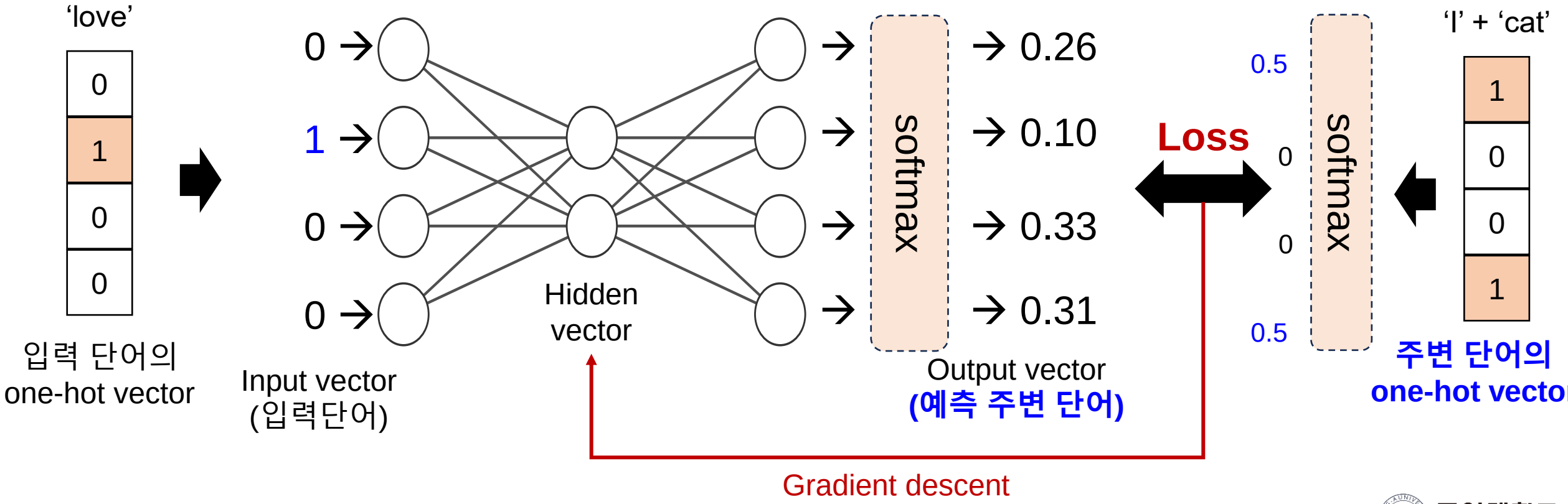




Word2Vec

- **Word2Vec 모델 학습 – Skip-Gram**
  - Ex. I love cat 문장 중 'love' 주변에 나올 수 있는 단어를 예측 하도록 학습





Questions & Answers

Dongsan Jun (dsjun@dau.ac.kr)
Image Signal Processing Laboratory (www.donga-ispl.kr)
Dong-A University, Busan, Rep. of Korea