

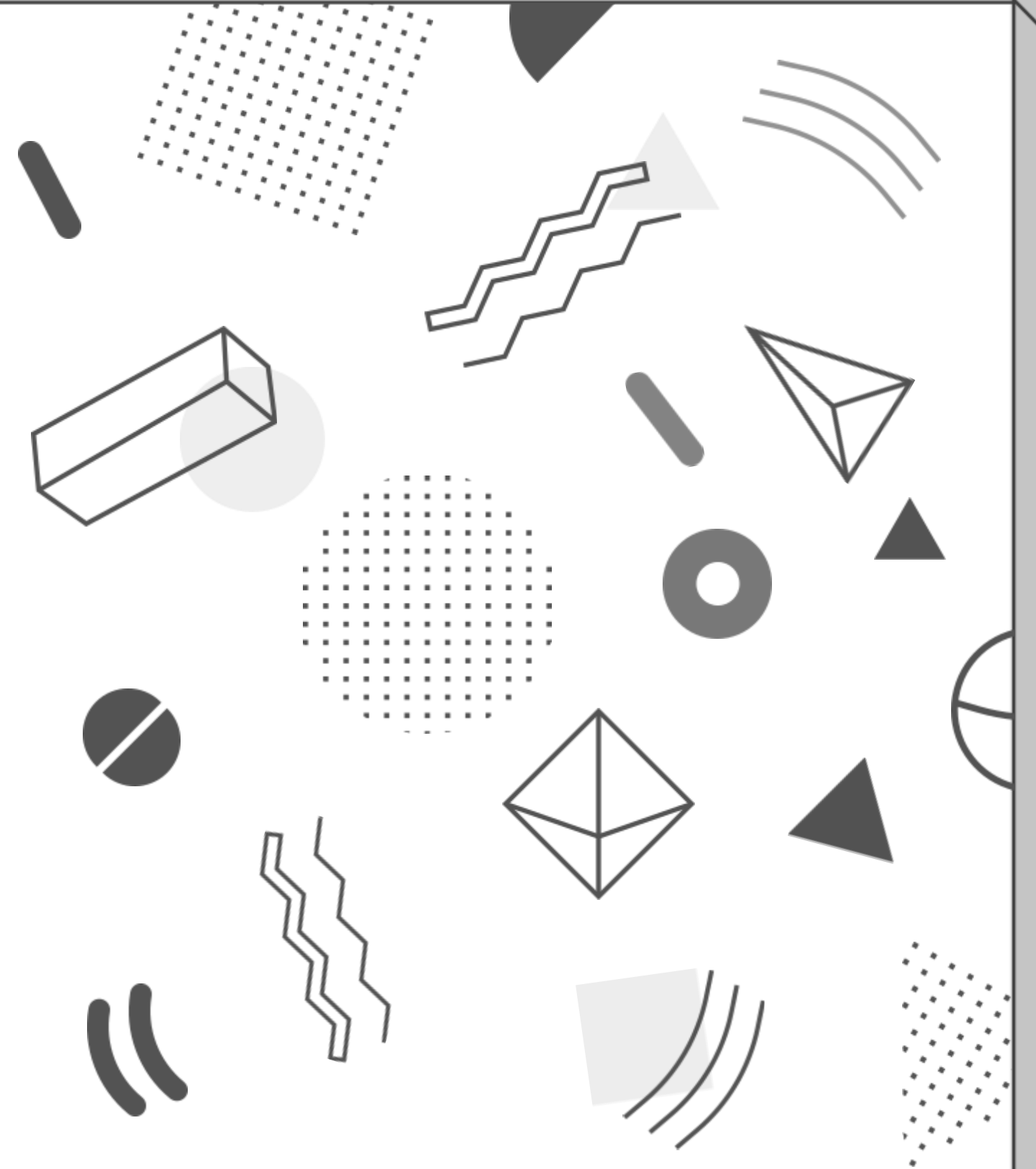


KUBIG 2022-2

# 여름방학 자연어처리(NLP) 분반

WEEK 3

분반장 : 14기 김유민  
분반원 : 15기 김지후, 김제성, 남정재, 이병주, 이승은



PART  
01



## 코드 구현과제

- 코드 구현과제 발표

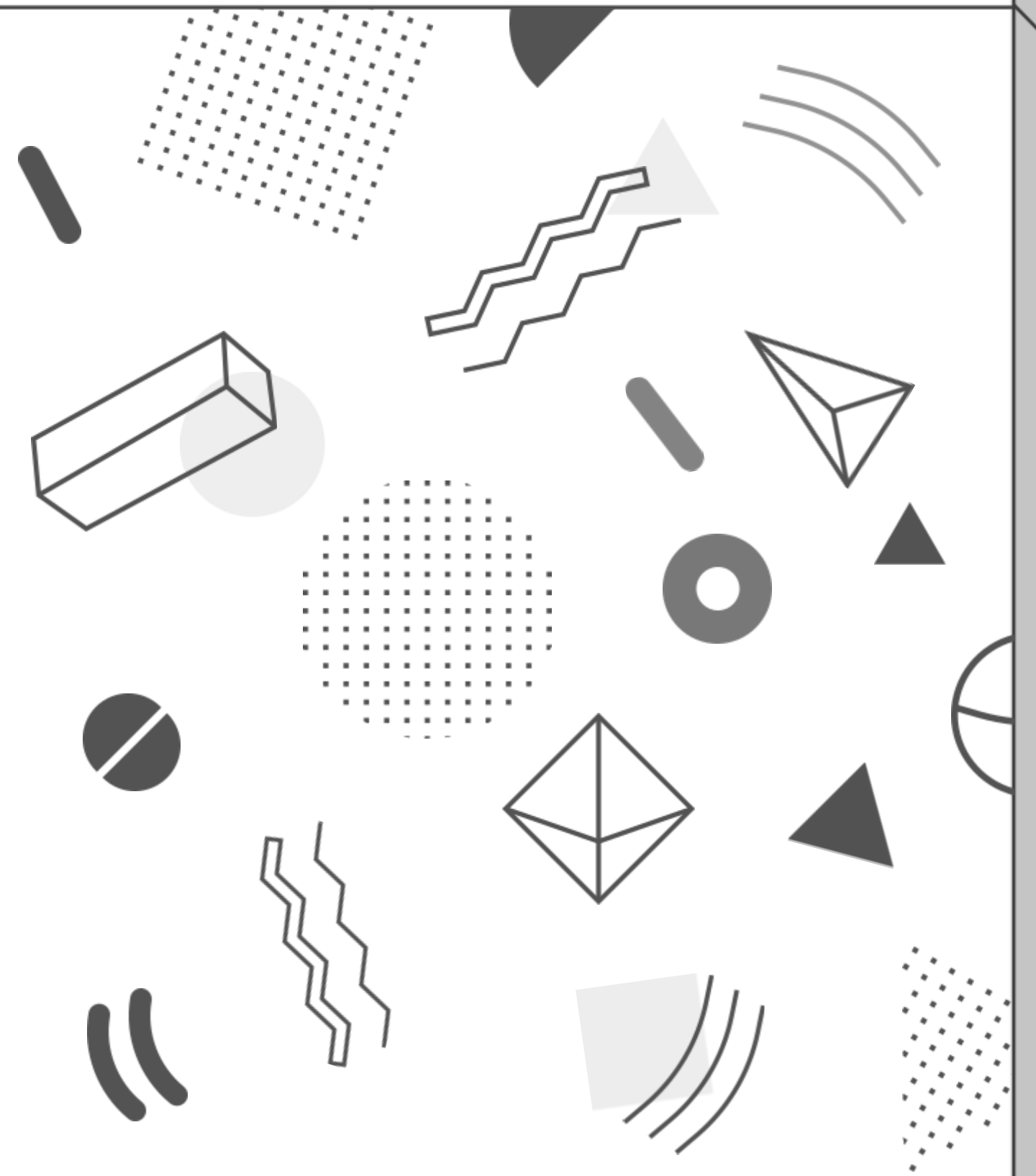
PART  
02



## 3주차 강의

- NNLM
- Word2Vec
- CBOW
- Skip-gram + Negative Sampling

# 01. 코드 구현과제



## 01. 코드 구현과제 | 과제 공유

---



- 돌아가면서 과제 해오신 코드 구현 설명 부탁드립니다!

+ ) 발표 포맷

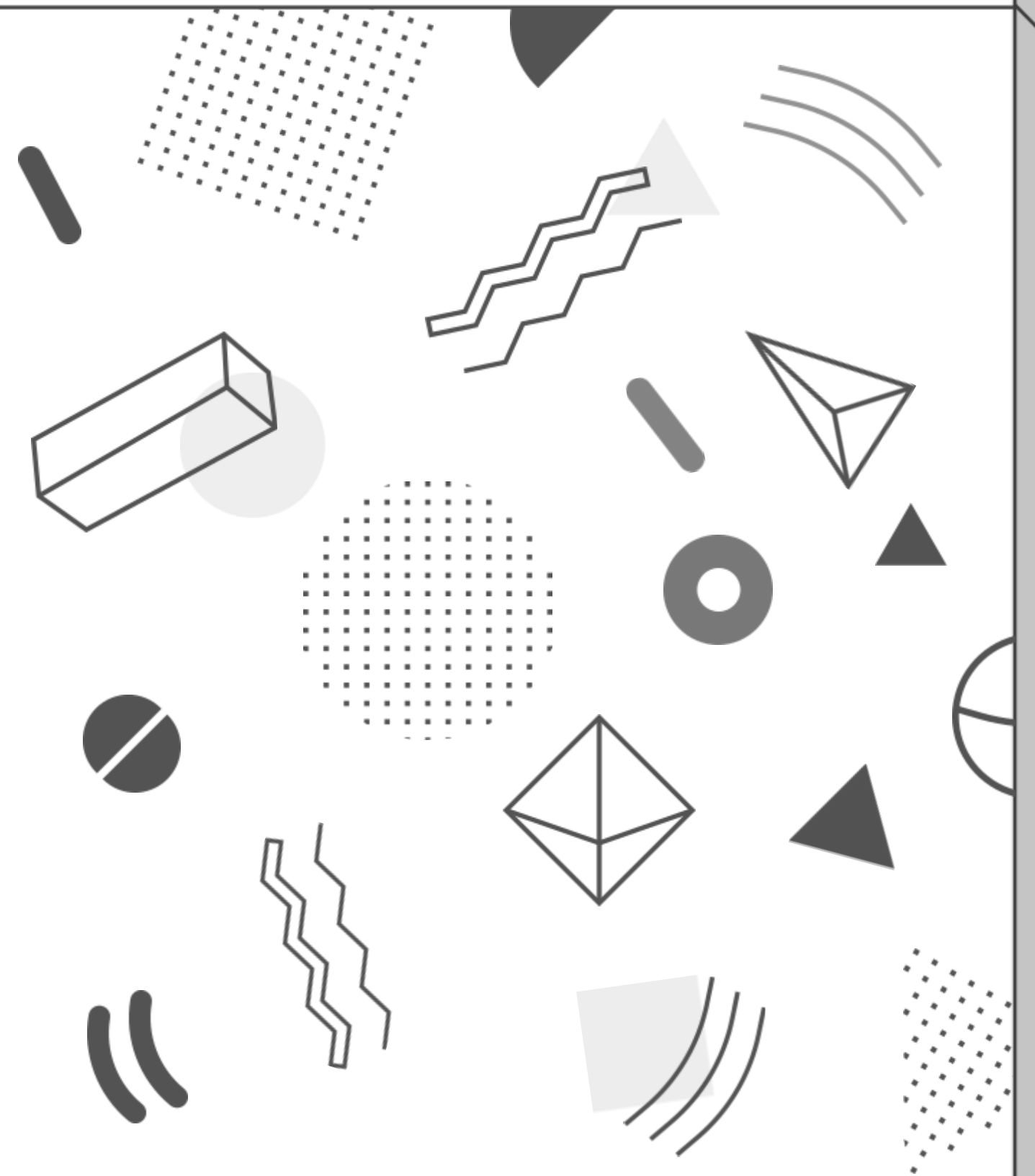
데이터셋 설명 -> 시도해본 3가지 이상 CASE -> 가장 적합한 코드 선택

최소 5분 최대 10분

## 02. 3주차 강의

- NNLM
- Word2Vec
- CBOW
- Skip-gram + Negative Sampling

노션 링크 : <https://onyx-tibia-876.notion.site/3-Word2Vec-a9a26f4760b042f6a103c015989040cc>



## 02. 2주차 강의

### N-gram 언어 모델

카운트를 기반으로 하며 희소 문제를 해결하기 위한 방법 중 하나  
단어의 확률을 구하고자 기준 단어의 앞 단어를 전부 포함해서 카운트하는 것이 아니라, 앞 단어 중 임의의 개수만 포함해서 카운트하여  
근사하자는 것

$$P(\text{is} | \text{An adorable little boy}) \approx P(\text{is} | \text{boy})$$

boy가 나왔을 때 is가 나올 확률

$$P(\text{is} | \text{An adorable little boy}) \approx P(\text{is} | \text{little boy})$$

little boy가 나왔을 때 is가 나올 확률

한계1. 여전히 희소 문제 존재

한계2. Trade-Off 문제

맥락을 파악하기 위해서는  $n$ 이 커질수록 good (근사의 정확도 ↑)

$n$ 이 커질수록 희소 문제의 심각성 ↑ (= 문장의 희소성 ↑)

⇒ 희소 문제의 근본적 해결 불가능 NLP 연구의 흐름이 인공 신경망 언어 모델로 이동하는 계기가 됨.



종류

unigrams : an, adorable, little, boy, is, spreading, smiles

bigrams : an adorable, adorable little, little boy, boy is, is spreading, spreading smiles

trigrams : an adorable little, adorable little boy, little boy is, boy is spreading, is spreading smiles

4-grams : an adorable little boy, adorable little boy is, little boy is spreading, boy is spreading smiles

ex)  $n = 3$

~~An adorable little~~ boy is spreading ?  
무시됨!  
n-1개의 단어

$$P(w | \text{boy is spreading}) = \frac{\text{count}(\text{boy is spreading } w)}{\text{count}(\text{boy is spreading})}$$

$$P(\text{insults} | \text{boy is spreading}) = 0.5$$

$$P(\text{smiles} | \text{boy is spreading}) = 0.2$$

⇒ insults 선택

## 02. 3주차 강의 | 워드 임베딩

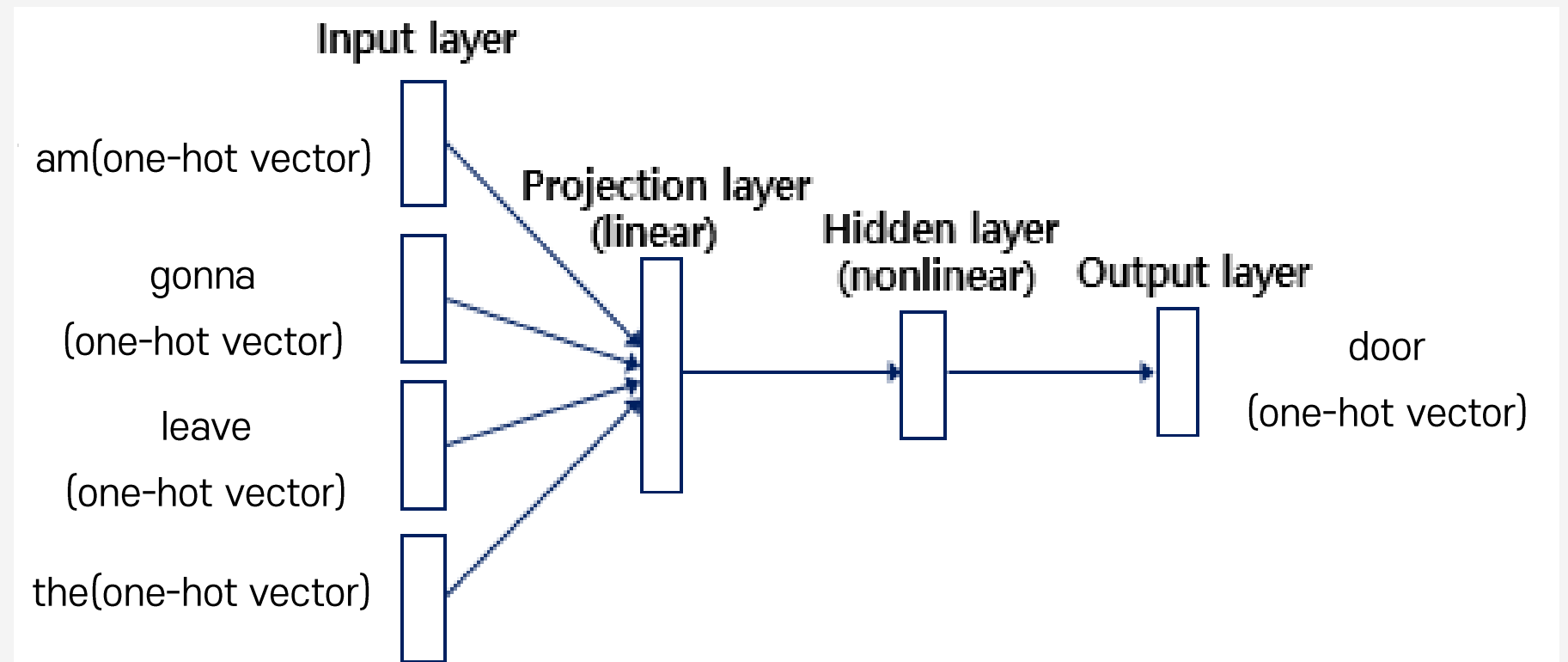
### 단어의 유사성

- Ex) 책을 꼼꼼히 읽어보다. 책을 정독하다
- 확률  $P(\text{정독하다}|\text{책을}) > \text{확률 } P(\text{요리하다}|\text{책을})$

### Neural Network Language Model (NNLM)

- I am gonna leave the door open
- I am gonna leave the -> ?

I = [1, 0, 0, 0, 0, 0, 0]  
am = [0, 1, 0, 0, 0, 0, 0]  
gonna = [0, 0, 1, 0, 0, 0, 0]  
leave = [0, 0, 0, 1, 0, 0, 0]  
the = [0, 0, 0, 0, 0, 1, 0]  
door = [0, 0, 0, 0, 0, 0, 1, 0]  
open = [0, 0, 0, 0, 0, 0, 1]

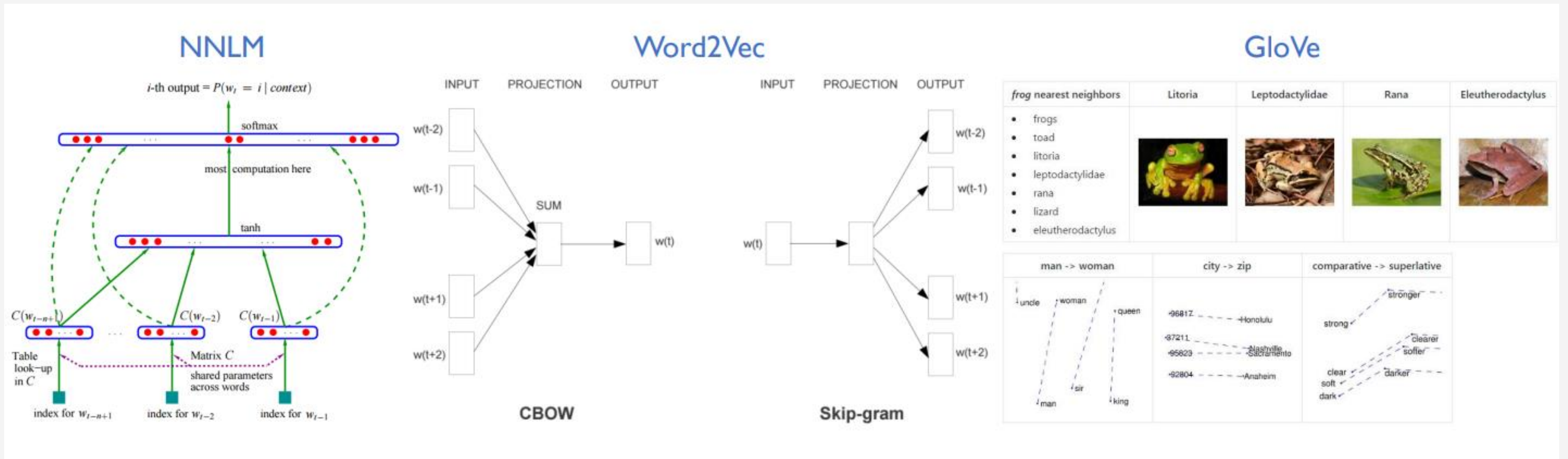


window = 4

## 02. 3주차 강의 | 워드 임베딩

### Word Embedding 워드 임베딩

- 텍스트를 컴퓨터가 이해하고, 효율적으로 처리하기 위해서는 컴퓨터가 이해할 수 있도록 텍스트를 적절한 숫자로 변환해야 하는데, 단어를 수치화 하기 위해 단어를 인공 신경망 학습을 통해 벡터화 하는 방법
- The purpose of word embedding is to map the words in a language into a vector space so that semantically similar words are located close to each other.





## 02. 3주차 강의 | 워드 임베딩

### Word vectors: one-hot vector

- The most simple & intuitive representation
  - 원-핫 인코딩으로 표현하고자 하는 단어의 인덱스의 값만 1이고, 나머지 인덱스에는 전부 0으로 표현되는 벡터 표현법
- ⇒ 벡터 또는 행렬의 값이 대부분 0으로 표현됨 (희소 표현\_sparse representation)
- ⇒ 원핫 벡터는 희소벡터

$$w^{aardvark} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, w^a = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, w^{at} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \dots w^{zebra} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

- vector representation을 만들 수 있지만, similarities 보존 x

motel [0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0] AND  
hotel [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0] = 0

$$(w^{hotel})^T w^{motel} = (w^{hotel})^T w^{cat} = 0$$

## 02. 3주차 강의 | 워드 임베딩

### Word vectors: 분산 표현

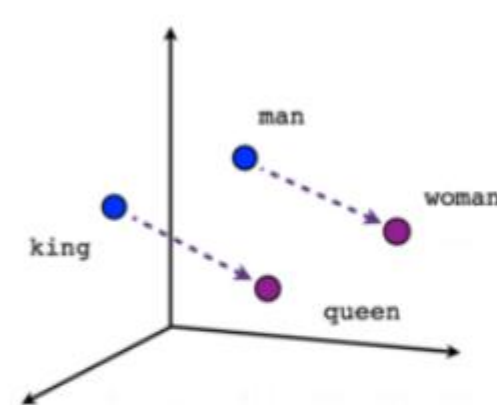
- A parameterized function mapping words in some language to a certain dimensional vectors
- 희소표현과 반대되는 표현으로, 벡터의 차원을 단어 집합의 크기로 상정하지 않고 사용자가 설정한 값으로 모든 단어의 벡터 표현의 차원을 맞춤
- ex) 워드 임베딩

$$W : \text{words} \rightarrow \mathbb{R}^n$$

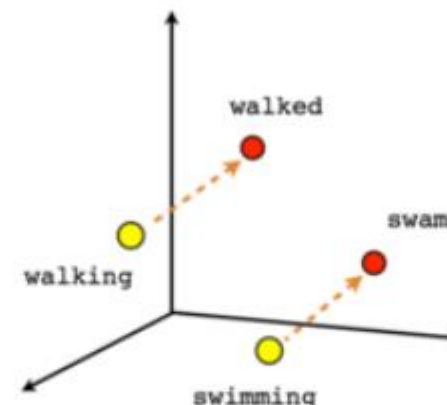
$$W(\text{"cat"}) = (0.2, -0.4, 0.7, \dots)$$

$$W(\text{"mat"}) = (0.0, 0.6, -0.1, \dots)$$

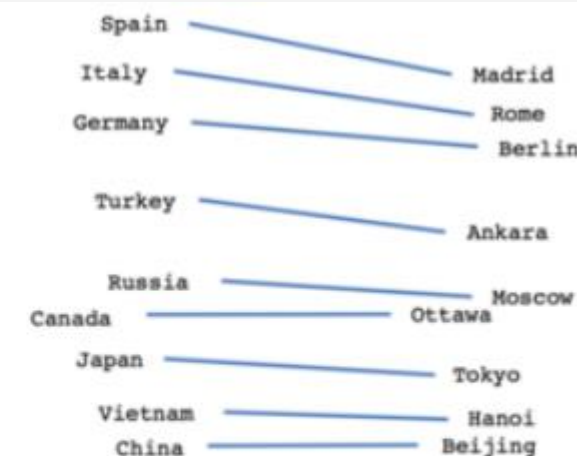
- ✓ Semantic relationship between words can be preserved



Male-Female



Verb tense

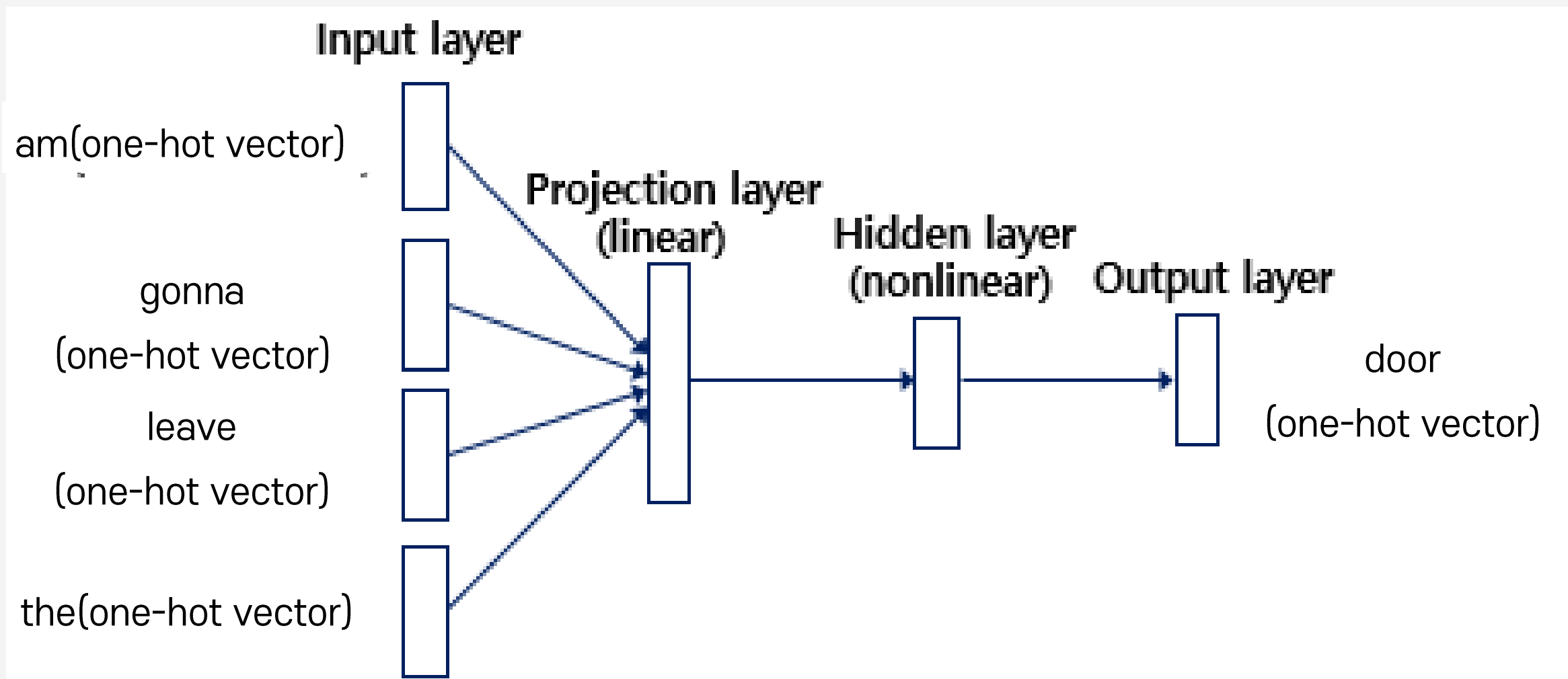


Country-Capital

## 02. 3주차 강의 | 워드 임베딩

### Neural Network Language Model (NNLM)

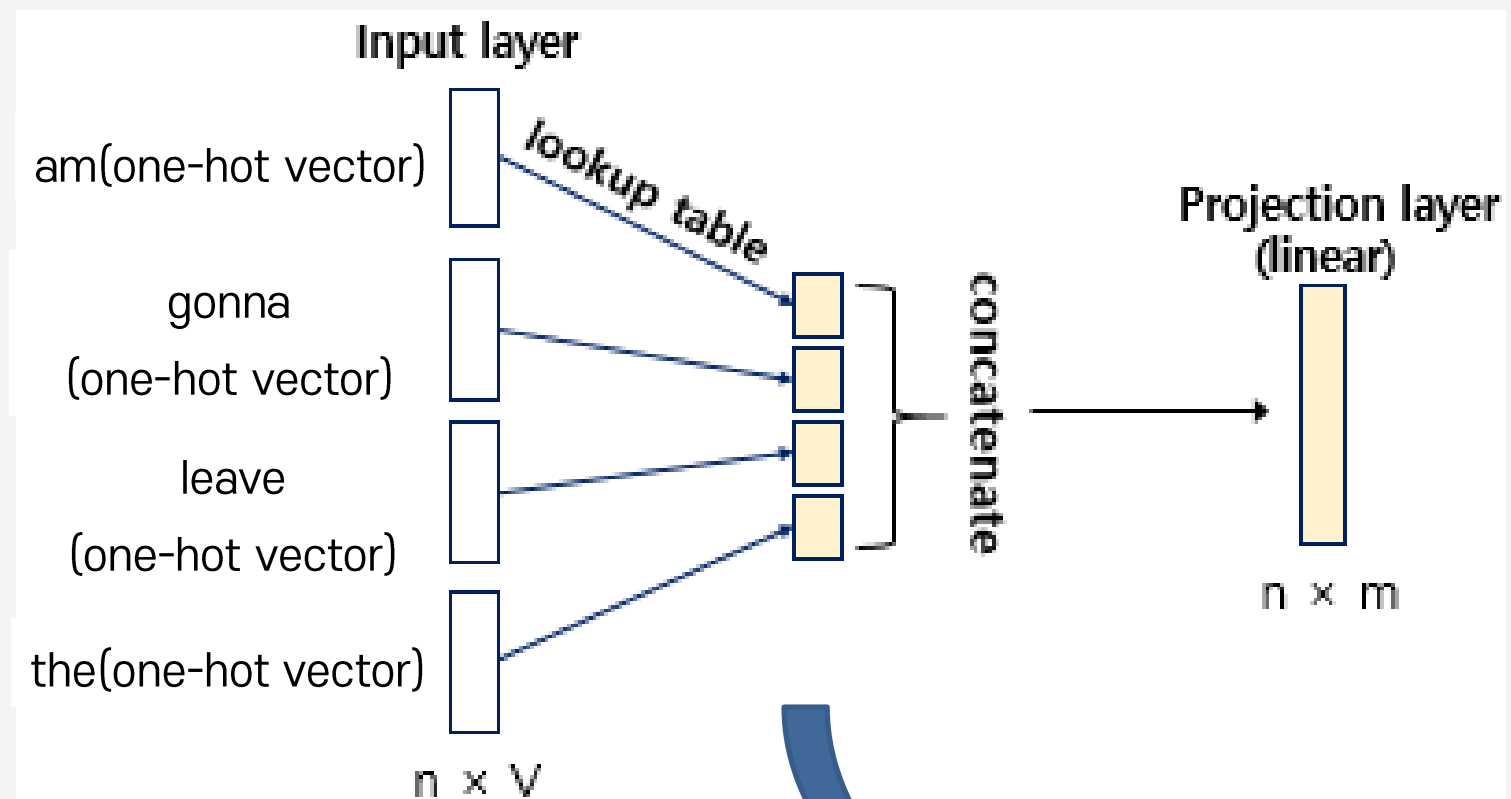
am = [0, 1, 0, 0, 0, 0, 0] <-



window = 4

## 02. 3주차 강의 | 워드 임베딩

### Neural Network Language Model (NNLM)



$$\mathbf{x}_{\text{leave}} \times \mathbf{W}_{V \times M} = \mathbf{e}_{\text{leave}}$$

The equation shows the calculation of the embedding vector  $\mathbf{e}_{\text{leave}}$  for the word "leave".

$\mathbf{x}_{\text{leave}}$  is a one-hot vector:  $[0, 0, 0, 1, 0, 0, 0]$ .

$\mathbf{W}_{V \times M}$  is the weight matrix (lookup table):

0.5	2.1	1.9	1.5	0.8
0.8	1.2	2.8	1.8	2.1
0.1	0.8	1.2	0.9	0.7
2.1	1.8	1.5	1.7	2.7

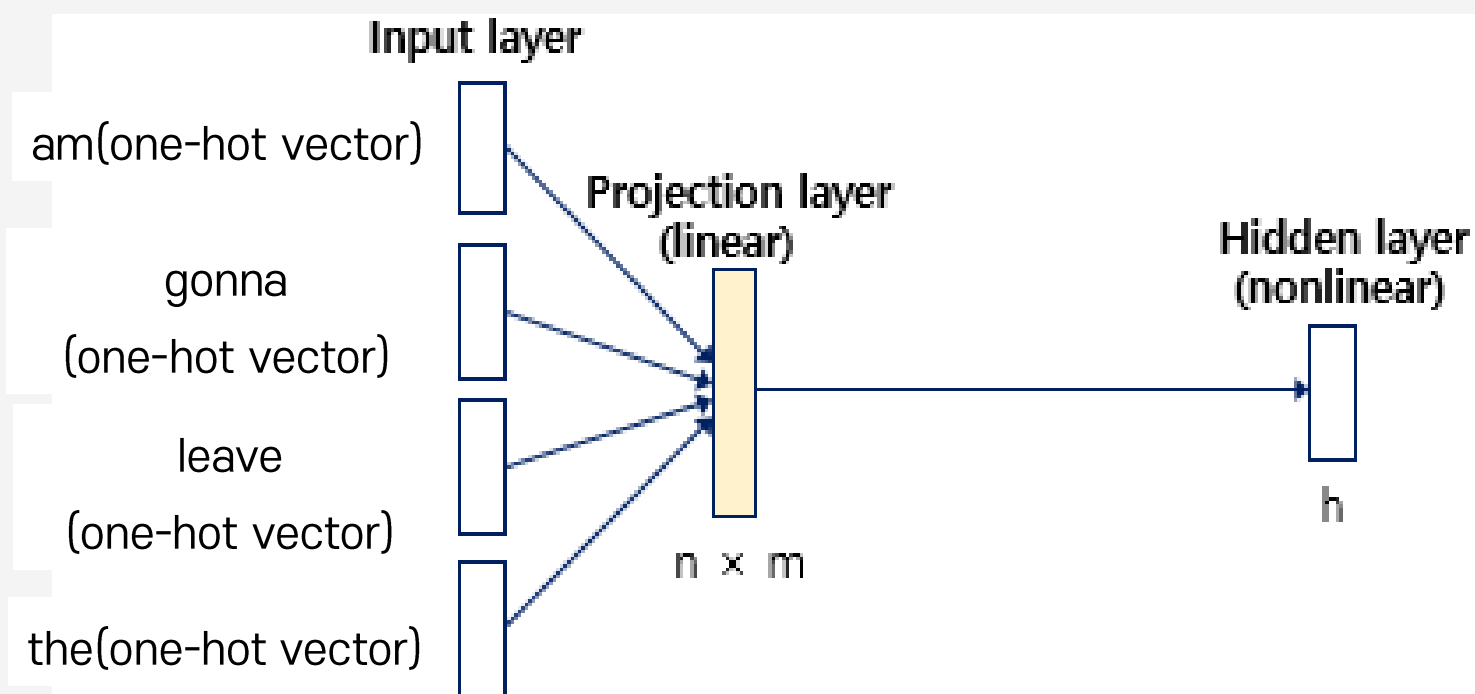
The result  $\mathbf{e}_{\text{leave}}$  is the embedding vector:  $[2.1, 1.8, 1.5, 1.7, 2.7]$ .

The lookup table is labeled **lookup table**.

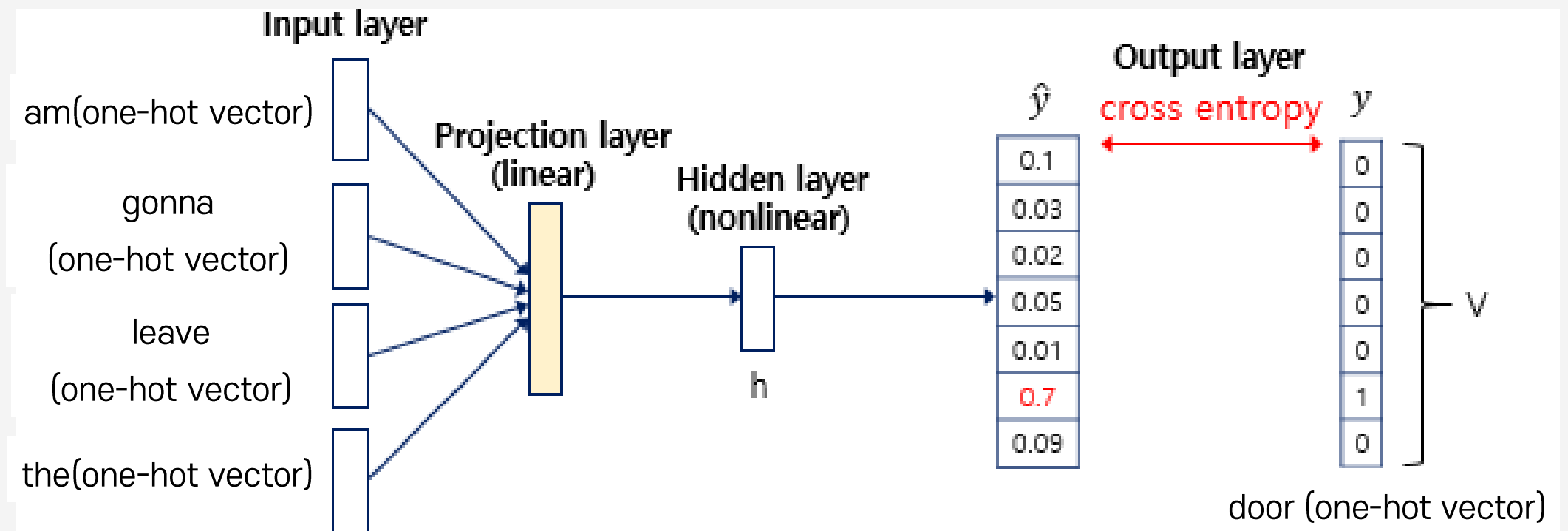
임베딩 벡터(embedding vector)

## 02. 3주차 강의 | 워드 임베딩

### Neural Network Language Model (NNLM)



$$\text{은닉층} : h^{layer} = \tanh(W_h p^{layer} + b_h)$$



$$\text{출력층} : \hat{y} = \text{softmax}(W_y h^{layer} + b_y)$$

- 0과 1사이의 실수값
- 총 합은 1

## 02. 3주차 강의 | 워드 임베딩

---

### Neural Network Language Model (NNLM)

- Represent words as dense vectors in  $\mathbb{R}^n$  (word embeddings).

$\mathbf{w}_t \in \mathbb{R}^{|\mathcal{V}|}$  : One-hot representation of word  $\in \mathcal{V}$  at time  $t$

$\Rightarrow \mathbf{x}_t = \mathbf{X}\mathbf{w}_t$  : Word embedding ( $\mathbf{X} \in \mathbb{R}^{n \times |\mathcal{V}|}, n < |\mathcal{V}|$ )

- Train a neural net that composes history to predict next word.

$$\begin{aligned} p(w_t = j | w_1, \dots, w_{t-1}) &= \frac{\exp(\mathbf{p}^j \cdot g(\mathbf{x}_1, \dots, \mathbf{x}_{t-1}) + q^j)}{\sum_{j' \in \mathcal{V}} \exp(\mathbf{p}^{j'} \cdot g(\mathbf{x}_1, \dots, \mathbf{x}_{t-1}) + q^{j'})} \\ &= \text{softmax}(\mathbf{P}g(\mathbf{x}_1, \dots, \mathbf{x}_{t-1}) + \mathbf{q}) \end{aligned}$$

$\mathbf{p}^j \in \mathbb{R}^m, q^j \in \mathbb{R}$  : Output word embedding/bias for word  $j \in \mathcal{V}$

$g$  : Composition function

## 02. 3주차 강의 | 워드 임베딩

### Word2Vec

#### 1) CBOW

<div>중심 단어</div> <div>↓</div> <div>주변 단어</div> <div>↘</div> <div>The fat cat sat on the mat</div>	<table><tr><th>중심 단어</th><th>주변 단어</th></tr><tr><td>[1, 0, 0, 0, 0, 0, 0]</td><td>[0, 1, 0, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0, 0]</td></tr><tr><td>[0, 1, 0, 0, 0, 0, 0]</td><td>[1, 0, 0, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0, 0], [0, 0, 0, 1, 0, 0, 0]</td></tr><tr><td>[0, 0, 1, 0, 0, 0, 0]</td><td>[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]</td></tr><tr><td>[0, 0, 0, 1, 0, 0, 0]</td><td>[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]</td></tr><tr><td>[0, 0, 0, 0, 1, 0, 0]</td><td>[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]</td></tr><tr><td>[0, 0, 0, 0, 0, 1, 0]</td><td>[0, 0, 0, 1, 0, 0, 0], [0, 0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 0, 0, 1]</td></tr><tr><td>[0, 0, 0, 0, 0, 0, 1]</td><td>[0, 0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 0, 0, 1]</td></tr></table>	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, 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, 1]	[0, 0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 0, 0, 1]
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, 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, 1]	[0, 0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 0, 0, 1]																

- 주변에 있는 단어들을 입력으로 하여 중간 단어를 예측하는 방법

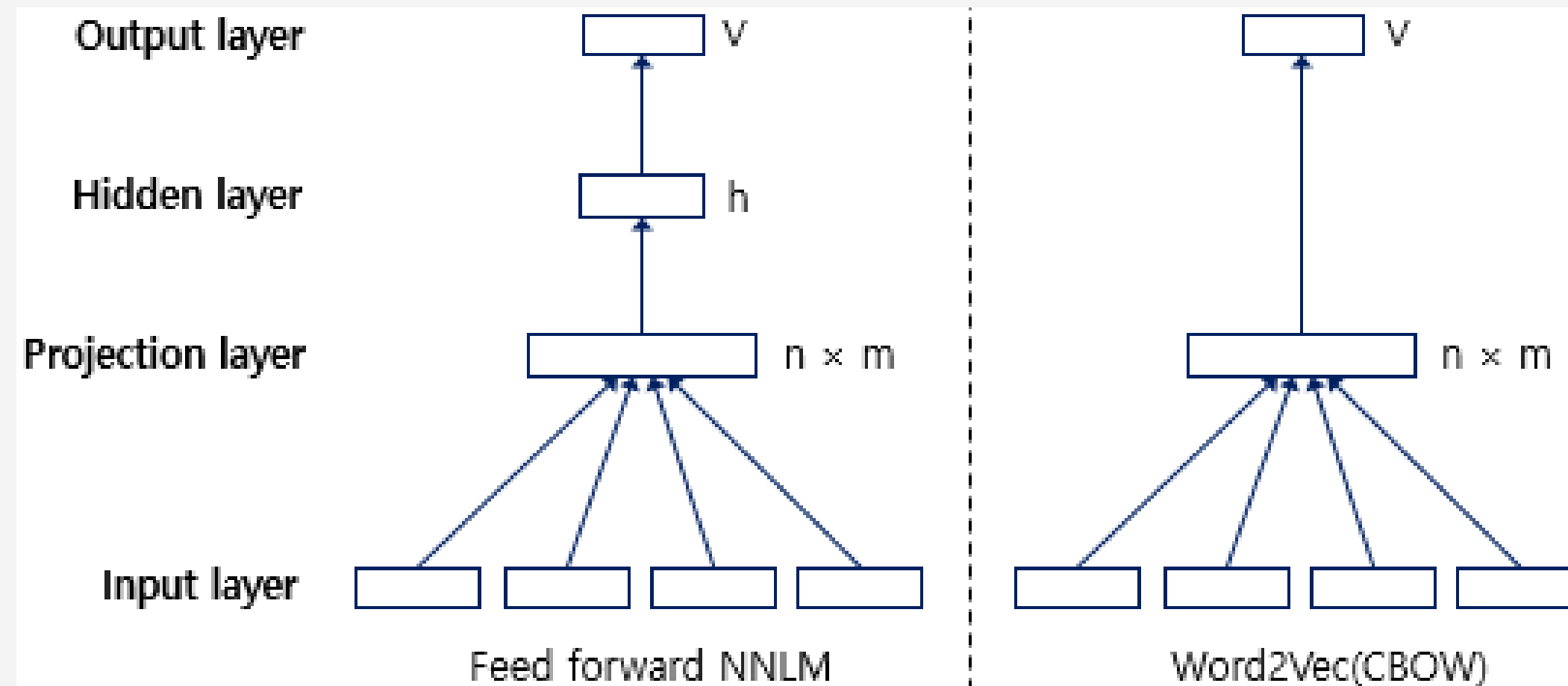
#### 2) skip-gram

<div>중심 단어</div> <div>↓</div> <div>주변 단어</div> <div>↘</div> <div>The fat cat sat on the mat</div>	<table><tr><th>중심 단어</th><th>주변 단어</th></tr><tr><td>cat</td><td>The</td></tr><tr><td>cat</td><td>Fat</td></tr><tr><td>cat</td><td>sat</td></tr><tr><td>cat</td><td>on</td></tr><tr><td>sat</td><td>fat</td></tr><tr><td>sat</td><td>cat</td></tr><tr><td>sat</td><td>on</td></tr><tr><td>sat</td><td>the</td></tr></table>	중심 단어	주변 단어	cat	The	cat	Fat	cat	sat	cat	on	sat	fat	sat	cat	sat	on	sat	the
중심 단어	주변 단어																		
cat	The																		
cat	Fat																		
cat	sat																		
cat	on																		
sat	fat																		
sat	cat																		
sat	on																		
sat	the																		

- 중간에 있는 단어들을 입력으로 주변 단어를 예측하는 방식 → 더 어려움

## 02. 3주차 강의 | 워드 임베딩

### NNLM Vs. Word2Vec



[예측 대상]

NNLM : 이전 단어로부터 다음 단어 예측

CBOW(Word2Vec) : 주변 단어로부터 중심 단어를 예측 / skip-gram : 중심단어로부터 주변 단어 예측

[구조]

NNLM : Input layer -> Projection layer -> Hidden layer -> Output layer

Word2Vec : Input layer -> Projection layer -> Output layer



## 02. 3주차 강의 | 워드 임베딩

### CBOW

중심 단어      주변 단어

↓      ↓

The fat cat sat on the mat

The fat cat sat on the mat

The fat cat sat on the mat

The fat cat sat on the mat

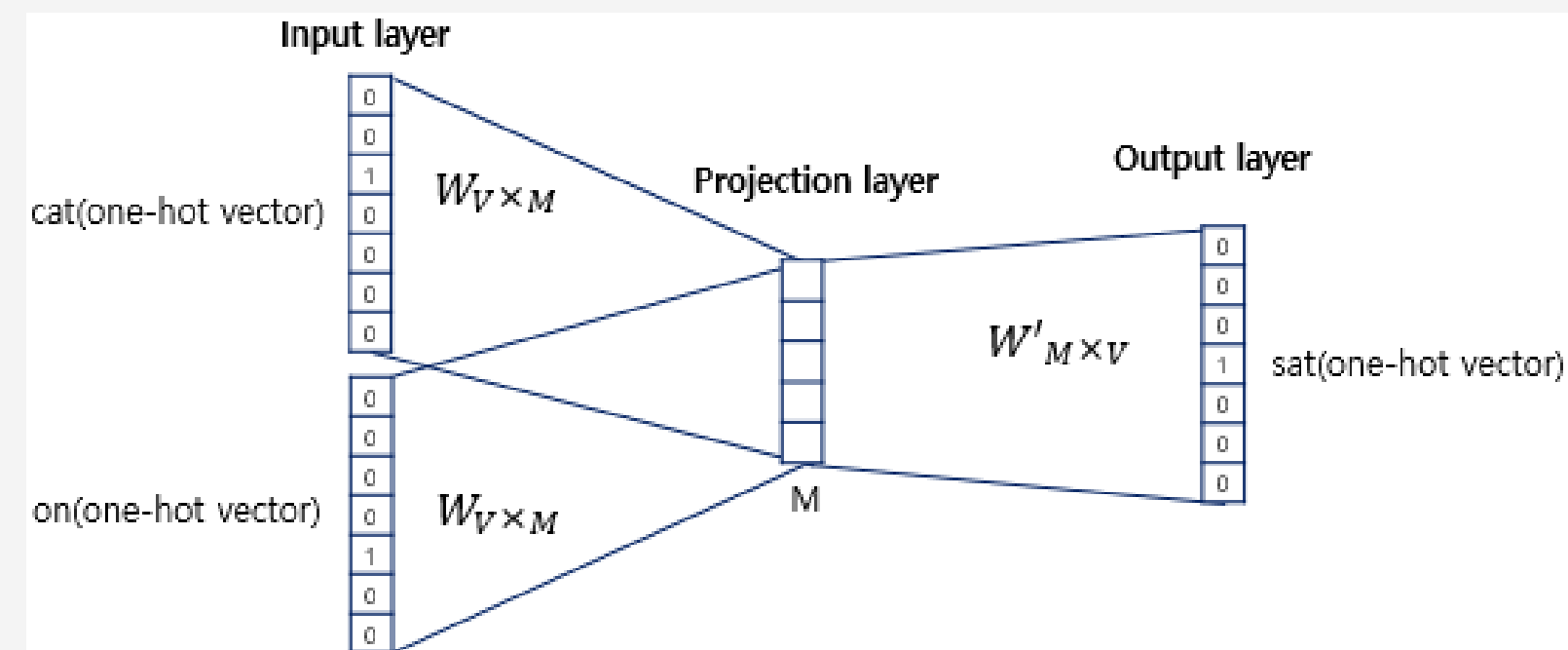
The fat cat sat on the mat

The fat cat sat on the mat

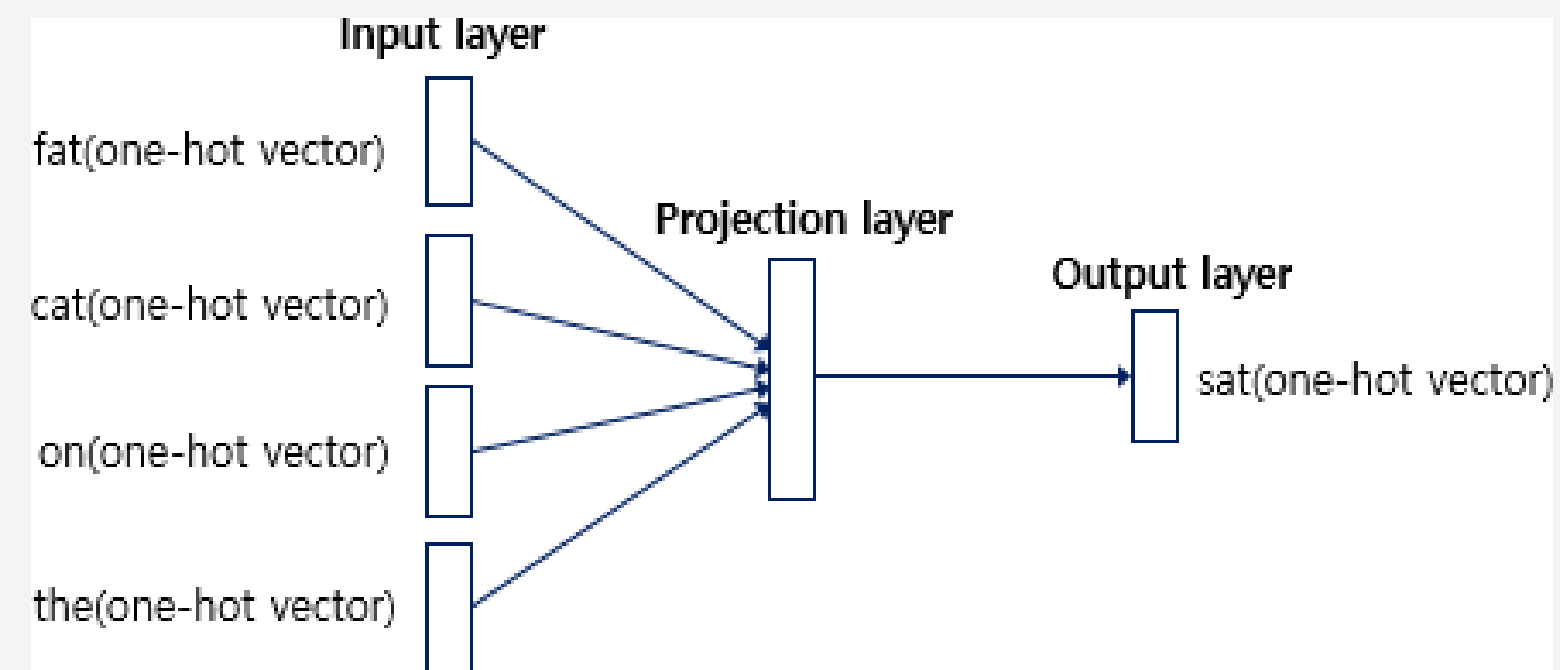
The fat cat sat on the mat

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, 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, 1]	[0, 0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 0, 1, 0]

- Window = 1 일 때



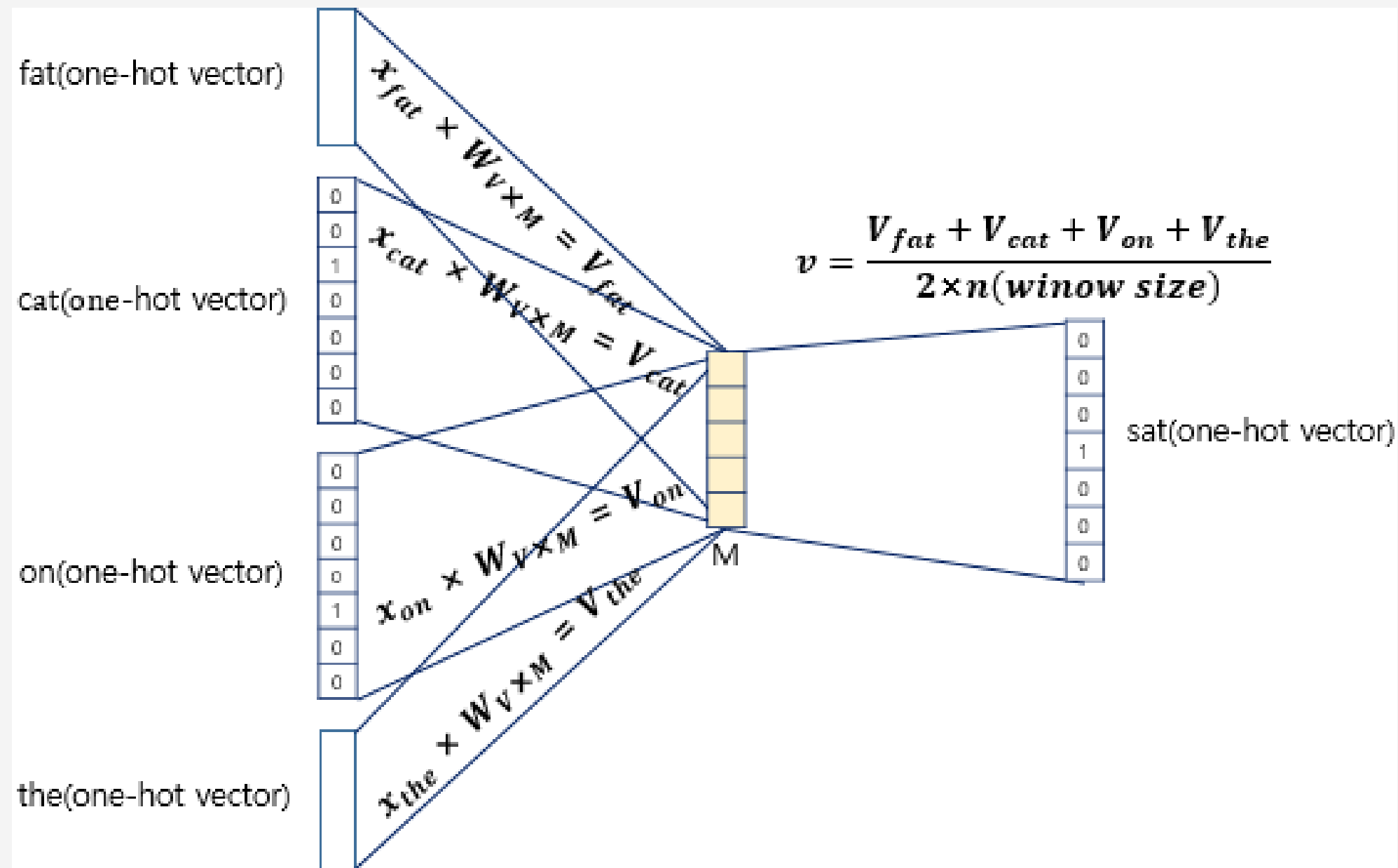
- Window = 2 일 때



## 02. 3주차 강의 | 워드 임베딩

### CBOW

- Window = 2 일 때



$$x_{cat} \times W_{V \times M} = V_{cat}$$

0	0	1	0	0	0	0
---	---	---	---	---	---	---

0.5	2.1	1.9	1.5	0.8
0.8	1.2	2.8	1.8	2.1
2.1	1.8	1.5	1.7	2.7

$$=$$

2.1	1.8	1.5	1.7	2.7
-----	-----	-----	-----	-----

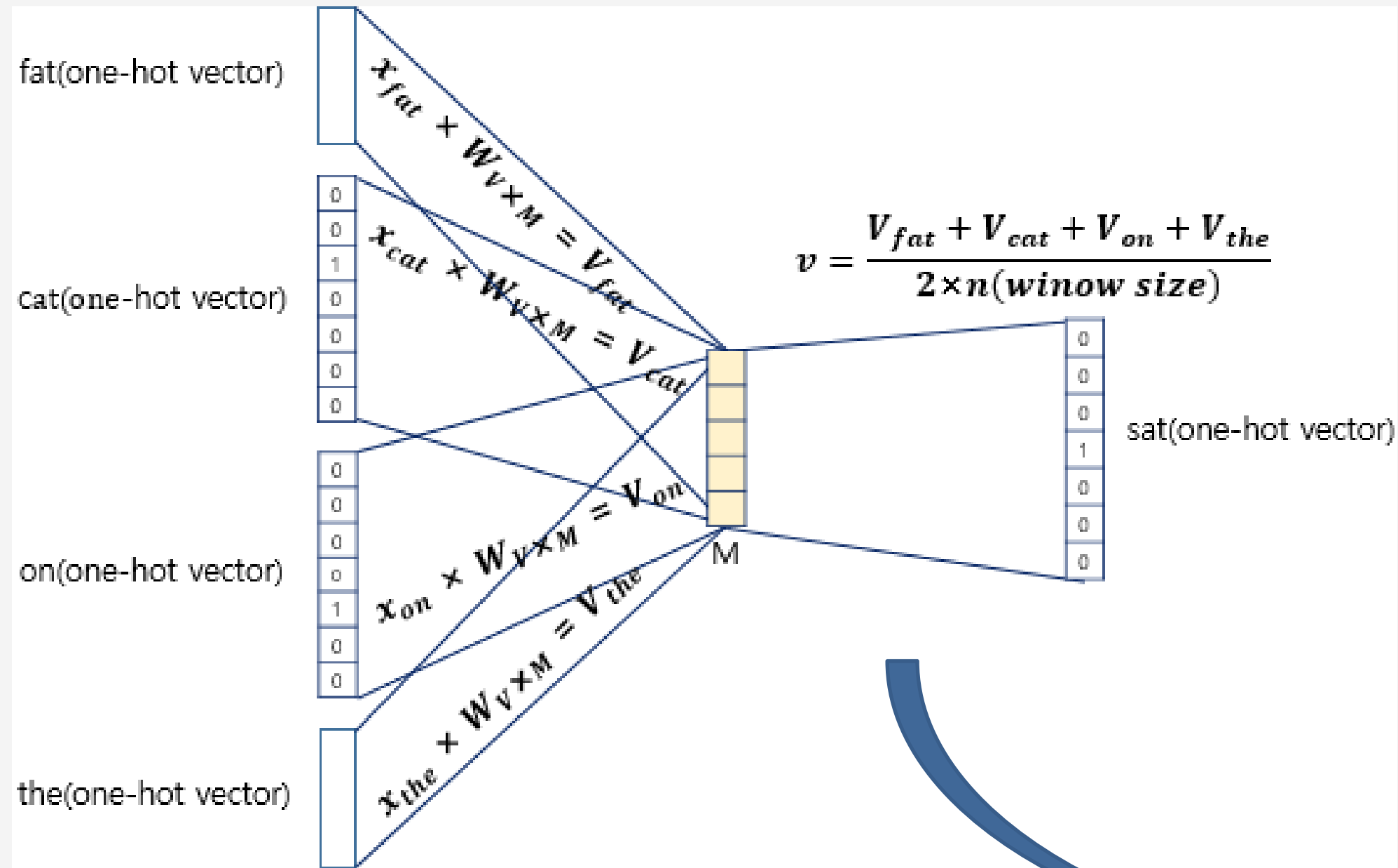
lookup table

window\_size = 2 일 때,  
input layer의 shape : [1,7] x 4  
W의 shape : [7,5]  
hidden\_layer의 shape : [1,5]  
⇒ W 행렬에서 window 인덱스들에 대한 행 평균이 필요  
W'의 shape : [5,7]  
output layer의 shape : [7,1]  
⇒ 입력 벡터 길이로 변환됨 → 각 단어의 score를 의미

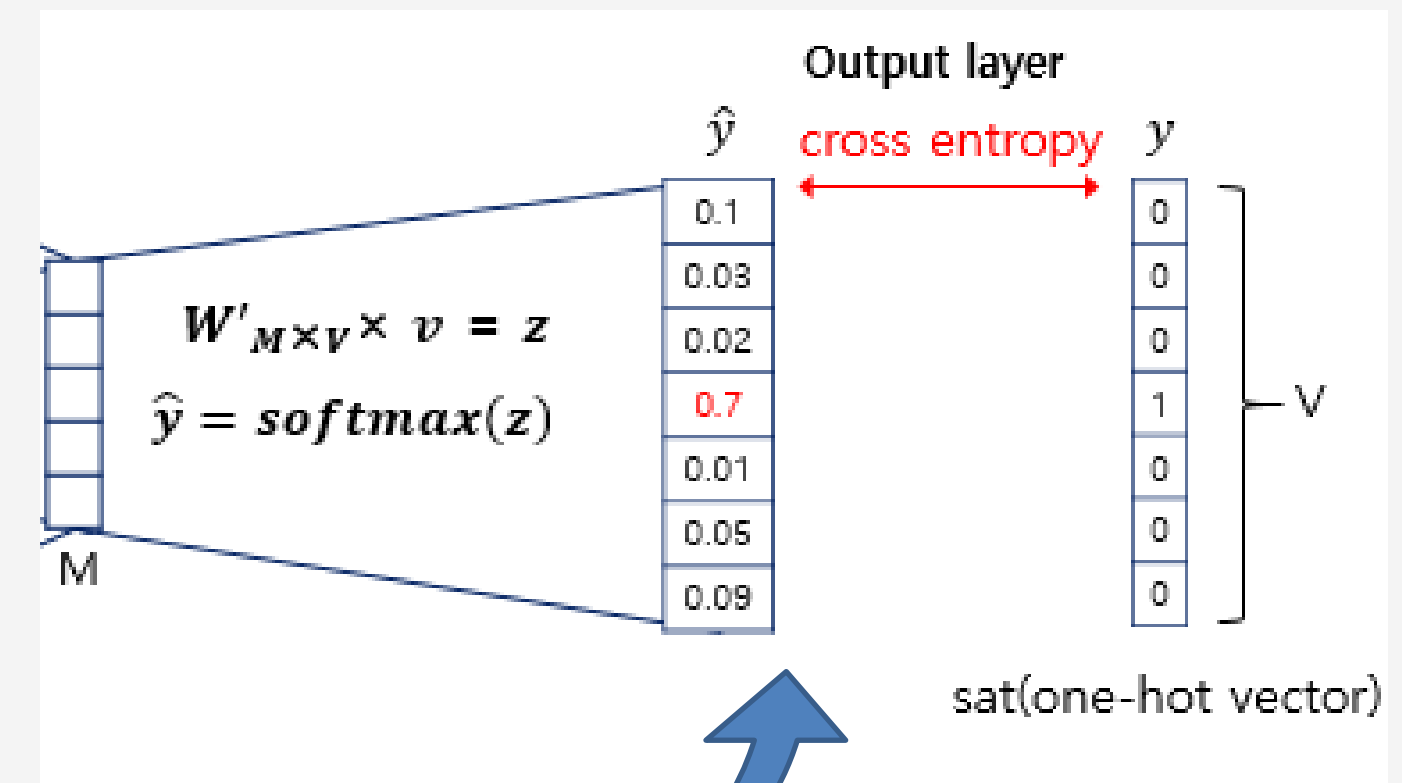
## 02. 3주차 강의 | 워드 임베딩

### CBOW

- Window = 2 일 때



$$\text{cost}(\hat{y}, y) = - \sum_{j=1}^V y_j \log(\hat{y}_j)$$

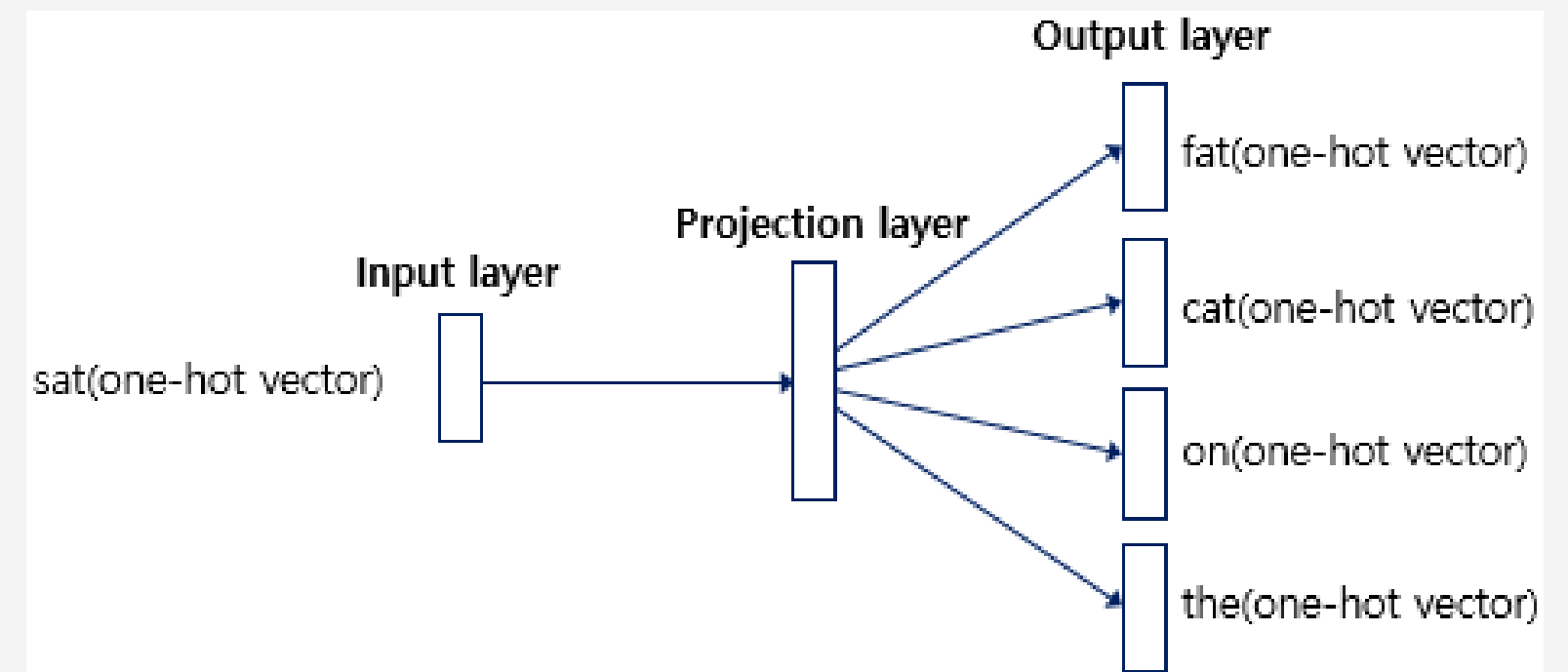


## 02. 3주차 강의 | 워드 임베딩

### Word2Vec : skip-gram

- 입력층에 중앙 단어(CBOW에서 타깃이었던 단어)로부터 주변의 여러 단어(맥락)를 추측
- 입력층이 하나고 출력층이 맥락의 수만큼 존재한다.
- Window =2 일 때

중심 단어	주변 단어
cat	The
cat	Fat
cat	sat
cat	on
sat	fat
sat	cat
sat	on
sat	the

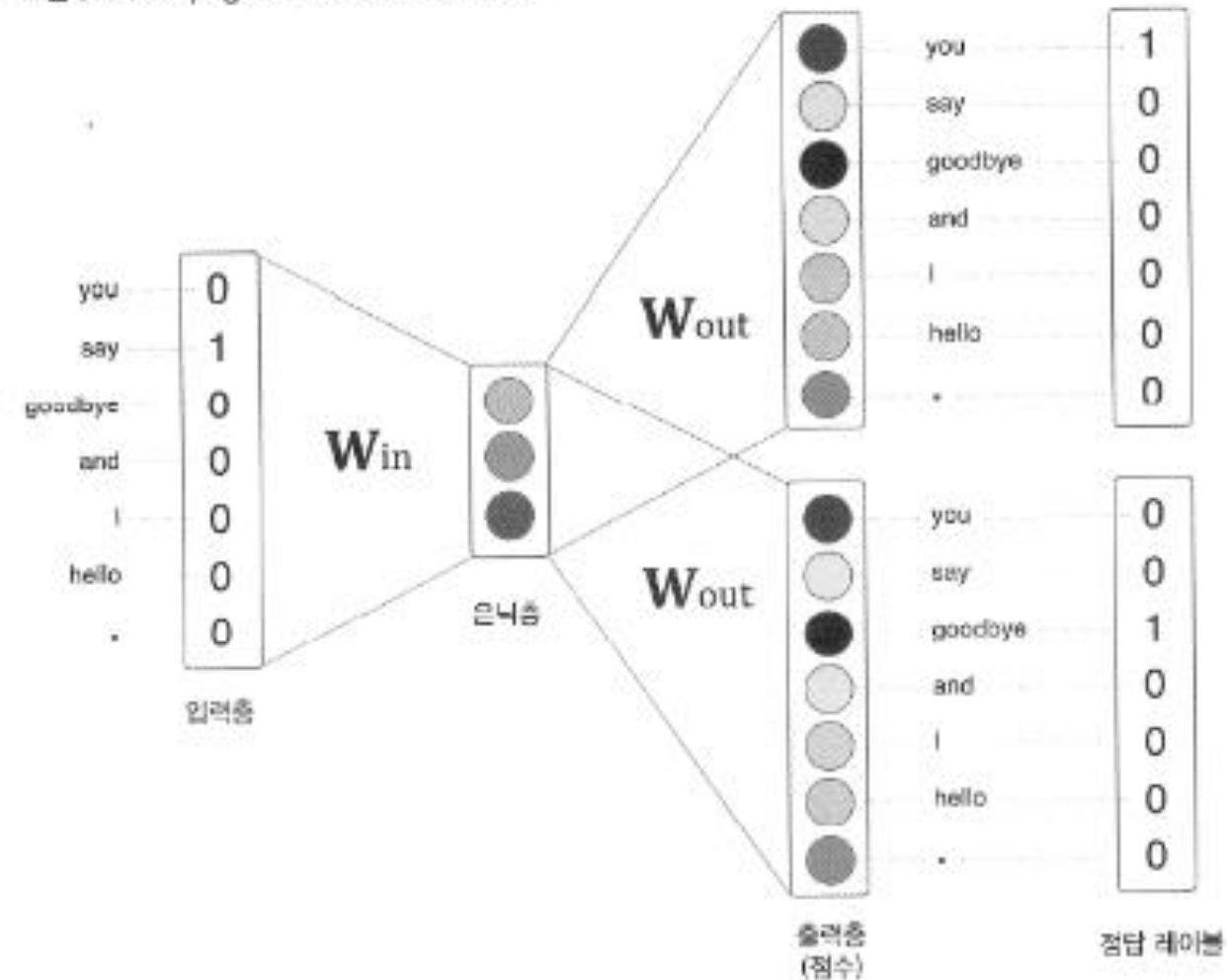


## 02. 3주차 강의 | 워드 임베딩

### Word2Vec : skip-gram

- \_\_\_\_ say \_\_\_\_ and I say hello.

그림 3-24 skip-gram 모델의 신경망 구성 예



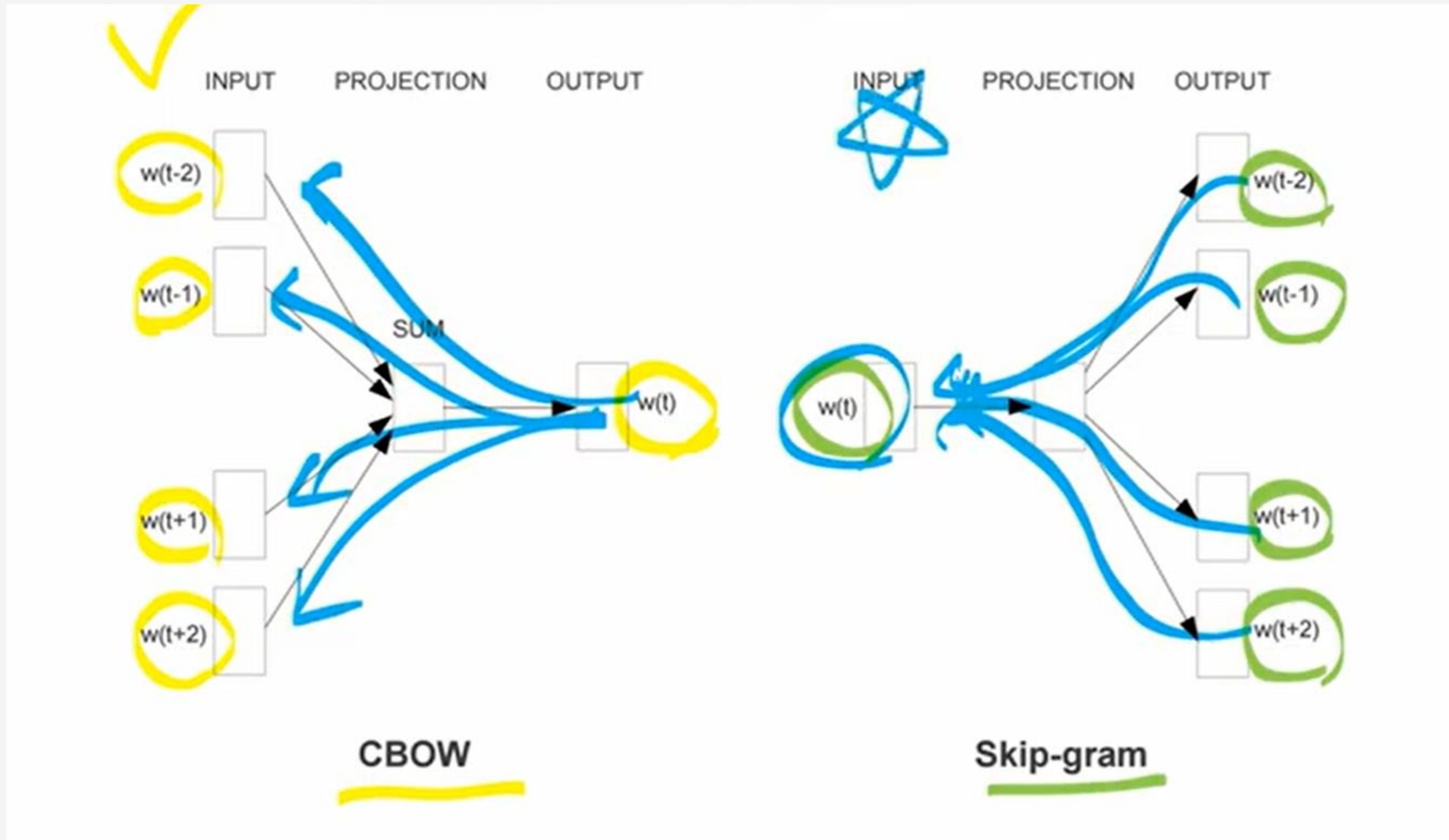
- **Objective function** : Maximize the log probability of any context word given the current center word

$$P(w_{t-1}, w_{t+1}|w_t) = P(w_{t-1}|w_t)P(w_{t+1}|w_t)$$

$$L = -\frac{1}{T} \sum_{t=1}^T \log P(w_{t-1}w_{t+1}|w_t) = -\frac{1}{T} \sum_{t=1}^T \log P(w_{t-1}|w_t)P(w_{t+1}|w_t)$$
$$= -\frac{1}{T} \sum_{t=1}^T (\log P(w_{t-1}|w_t) + \log P(w_{t+1}|w_t))$$

## 02. 3주차 강의 | 워드 임베딩

### Continuous bag-of-words (CBOW) vs. Skip-gram



## 02. 3주차 강의 | 워드 임베딩

### Skip-gram

- The number of weights to be trained:  $2 \times V \times N$  (Huge network!)

- ✓ Word pairs and phrases

- Treating common word pairs or phrases as single “word”

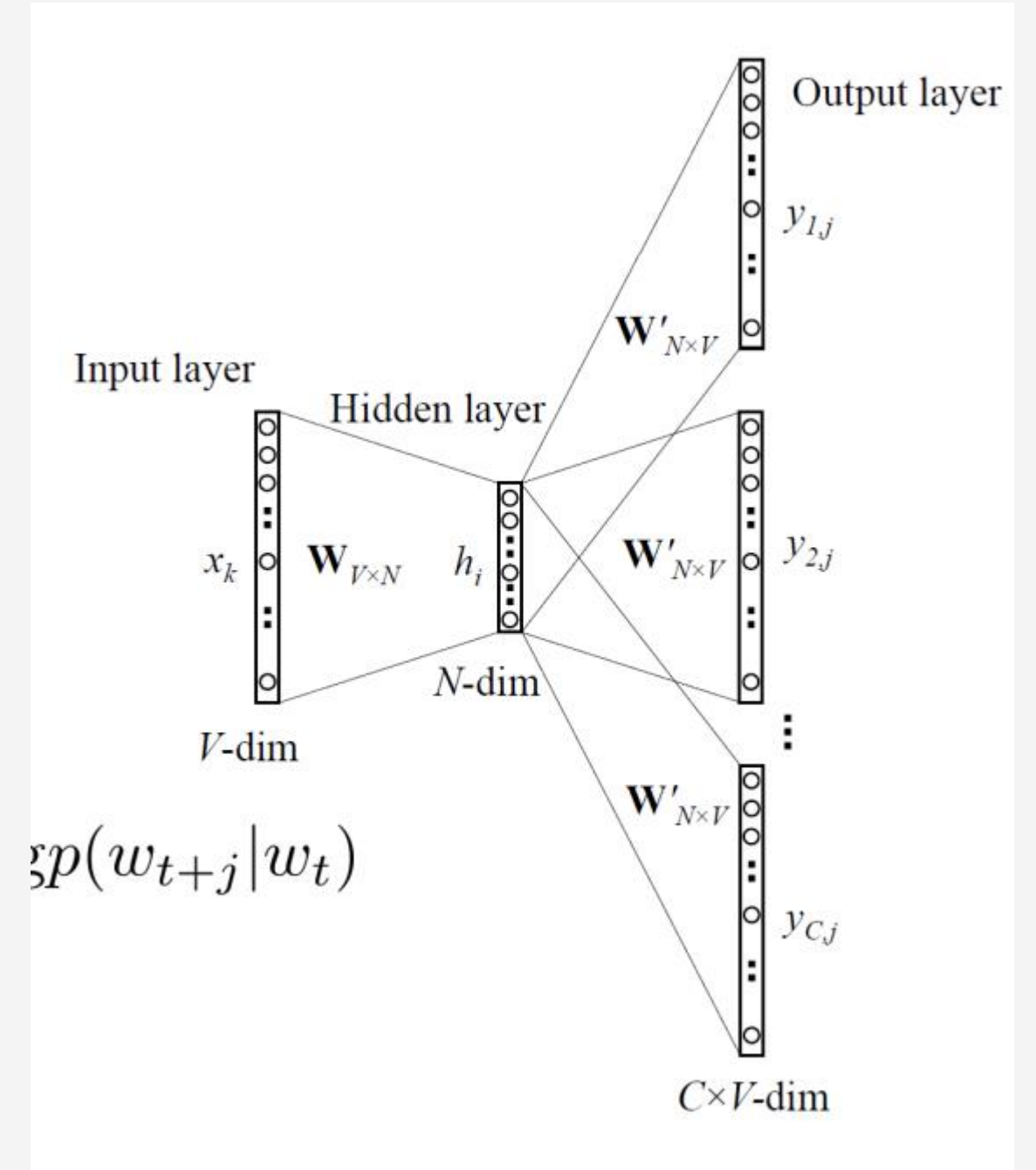
- ✓ Subsampling frequent words

- To decrease the number of training examples
- The probability of word  $w_i$  being removed

$$P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}} \quad \begin{array}{l} \leftarrow \text{Threshold } (10^{-5}) \\ \leftarrow \text{Term frequency in the corpus} \end{array}$$

$$\text{if } f(w_i) = 10^{-4}, P(w_i) = 1 - \sqrt{\frac{1}{10}} = 0.6838$$

$$\text{if } f(w_i) = 10^{-2}, P(w_i) = 1 - \sqrt{\frac{1}{1000}} = 0.9684$$



## 02. 3주차 강의 | 워드 임베딩

---

### Skip-gram

- The number of weights to be trained:  $2 \times V \times N$  (Huge network!)

- ✓ Negative sampling

- Instead of updating the weights associated with all output words, update the weight of a few (5-20) words

$$J(\theta) = \frac{1}{T} \sum_{t=1}^T \sum_{-m \leq j \leq m, j \neq 0} \log p(w_{t+j} | w_t) = \frac{1}{T} \sum_{t=1}^T J_t(\theta)$$

$$J_t(\theta) = \log \sigma(u_o^T v_c) + \sum_{i=1}^k E_{i \sim P(w)} [\log \sigma(-u_i^T v_c)]$$

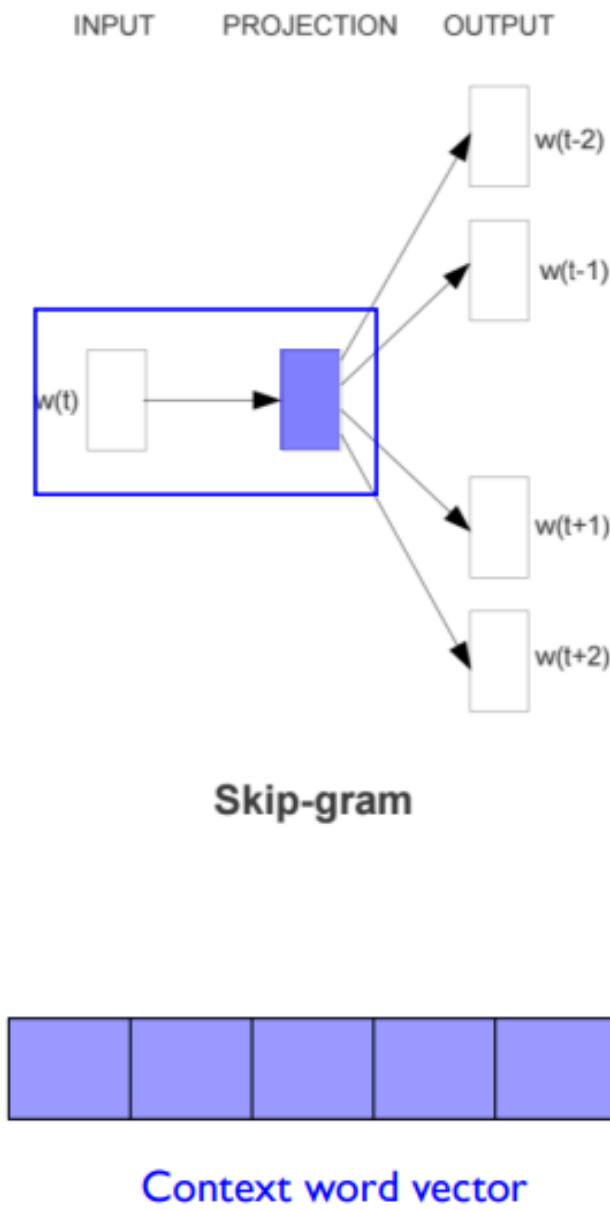
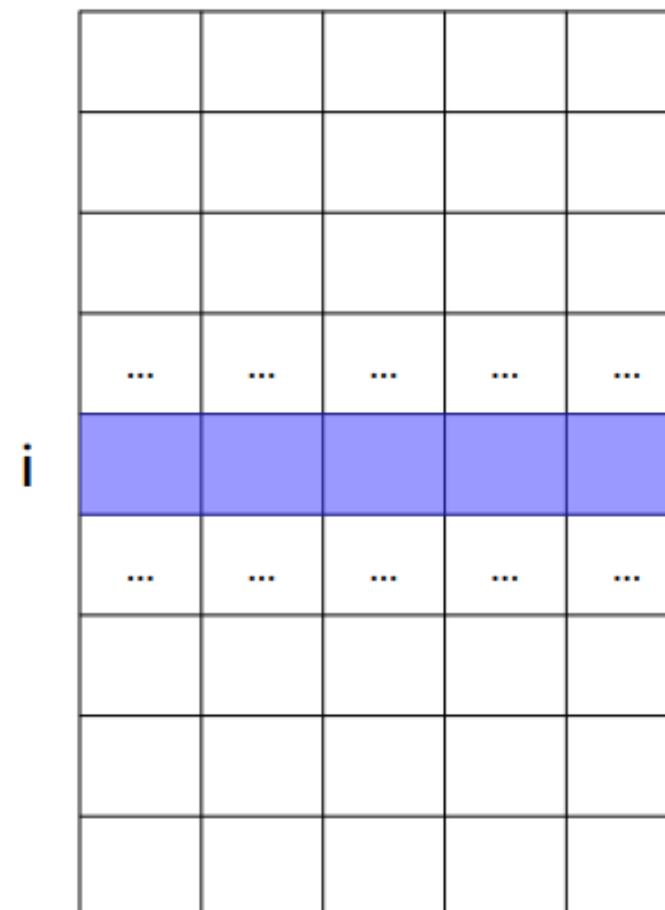
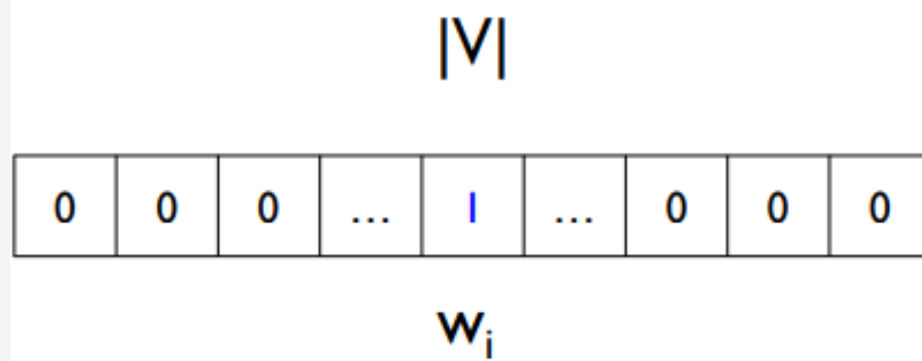
$$P(w_i) = \frac{f(w_i)^{3/4}}{\sum_{j=0}^n (f(w_j)^{3/4})}$$



## 02. 3주차 강의 | 워드 임베딩

### Negative sampling

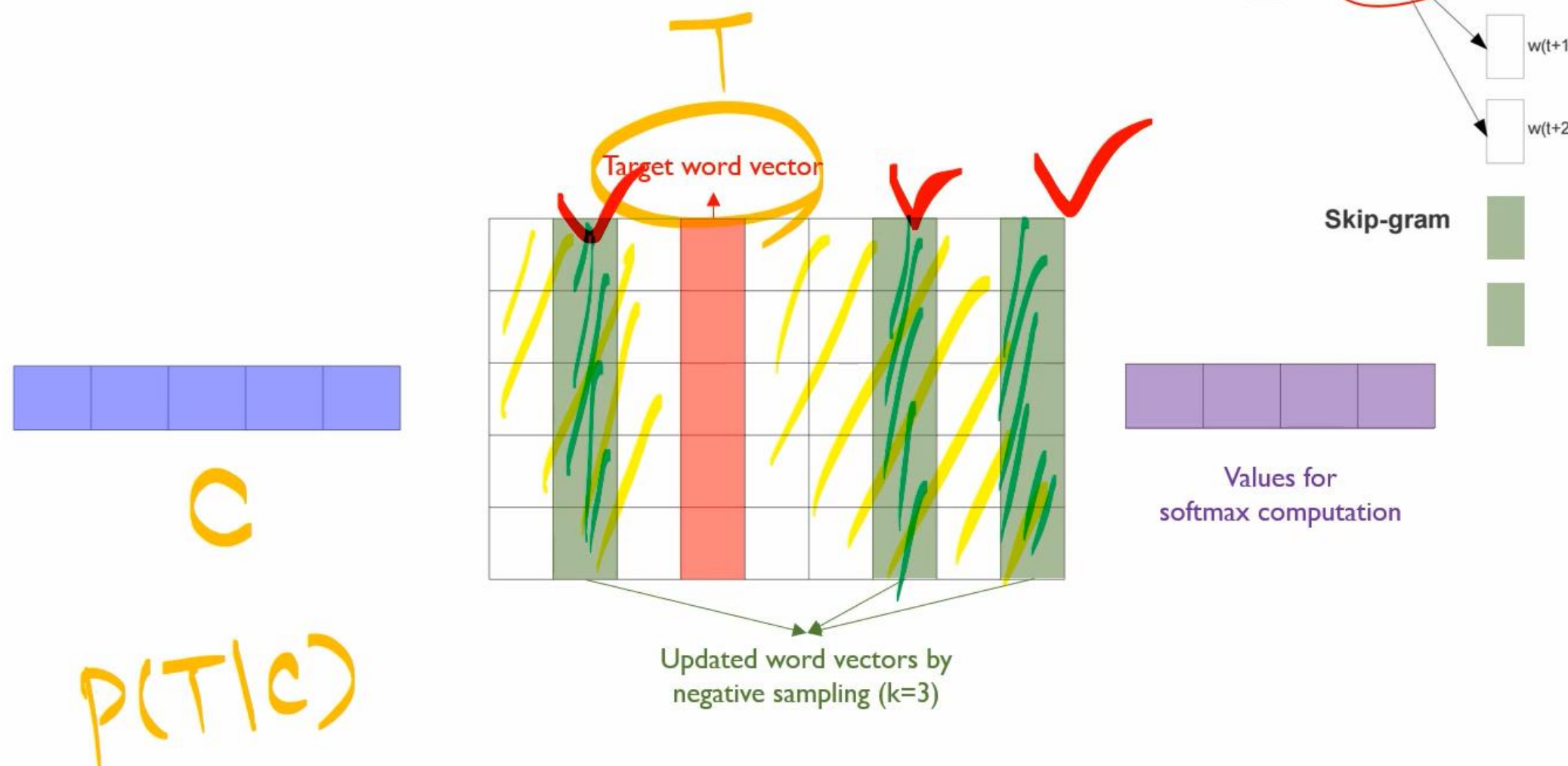
- Negative Sampling Example



## 02. 3주차 강의 | 워드 임베딩

### Negative sampling

- Negative Sampling Example
  - ✓ No. of 0 values for softmax computation is reduced from  $|V|$  to  $(k+1)$



## 02. 3주차 강의 | 워드 임베딩

### Negative sampling

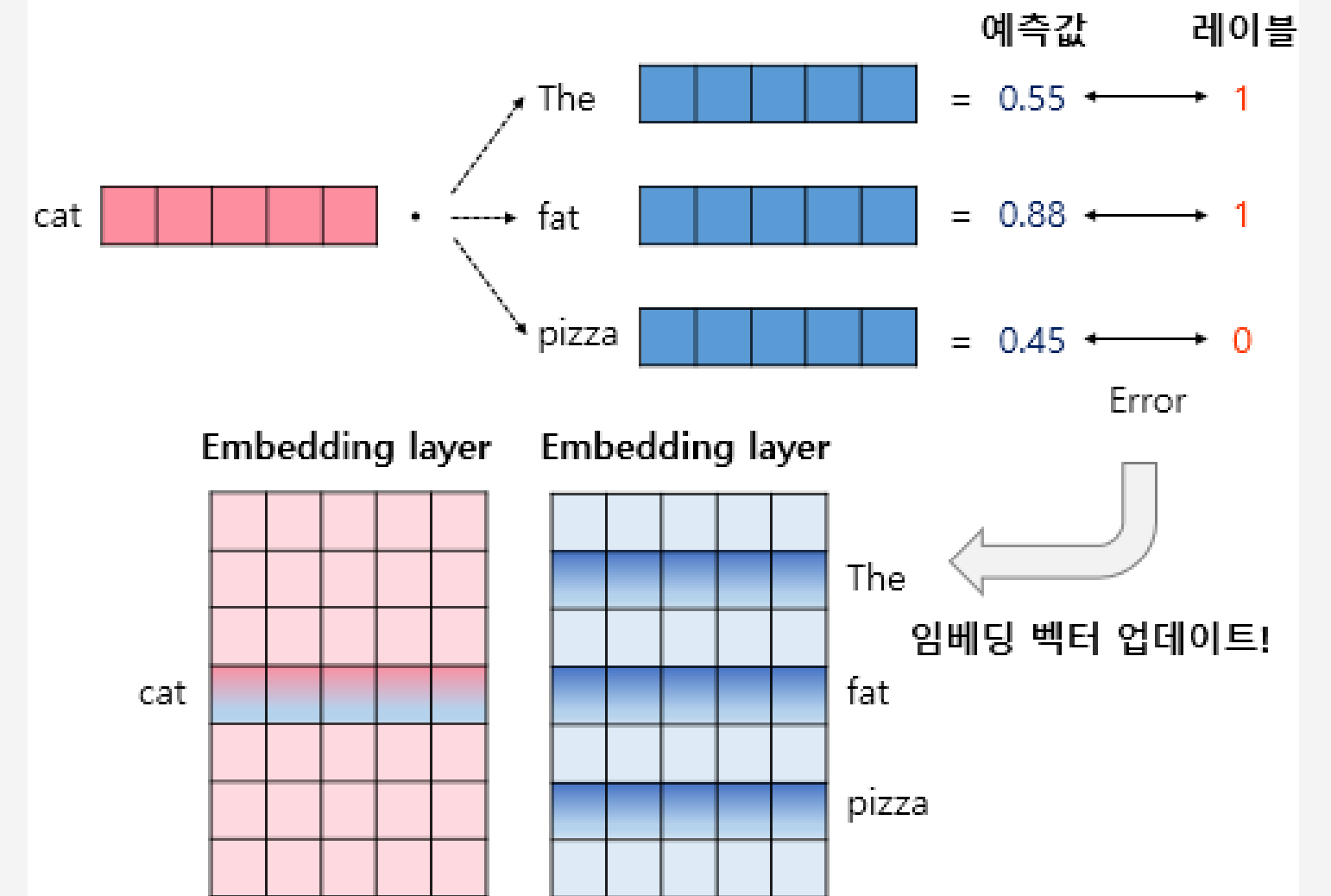
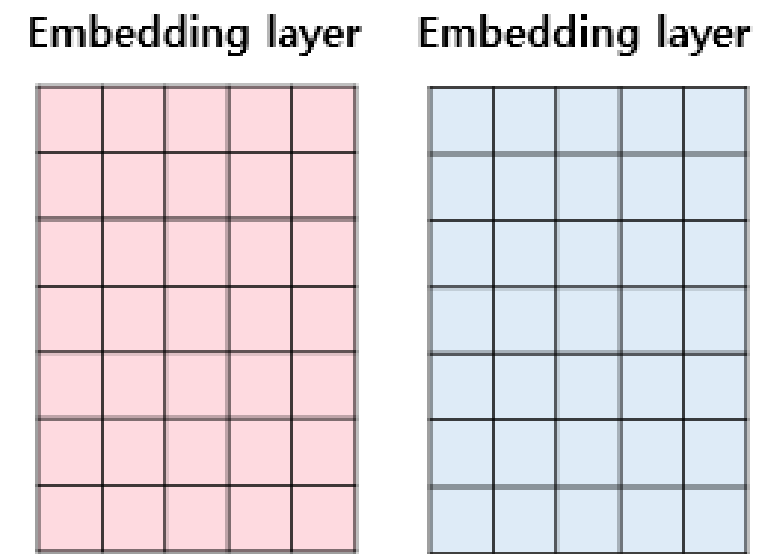
- softmax 함수의 연산량 증가  $\Rightarrow$  네거티브 샘플링을 통해 해결
- 이진 분류를 활용하여 정답에 대한 학습 뿐만 아니라 오답에 대해서도 학습하여 추론의 정확성을 높이는 방식
- What 보다는 Yes/No의 질문으로 변환  $\Rightarrow$  계산량 증가를 막음



## 02. 3주차 강의 | 워드 임베딩

### Negative sampling

입력1	입력2	레이블
cat	The	1
cat	fat	1
cat	pizza	0
cat	computer	0
cat	sat	1
cat	on	1
cat	cute	1
cat	mighty	0
...	...	...



- 중심 단어와 주변 단어의 내적값을 이 모델의 예측값으로 하고, 레이블과의 오차로부터 역전파하여 중심 단어와 주변 단어의 임베딩 벡터값을 업데이트합니다.

## 02. 3주차 강의 | 워드 임베딩

### Negative sampling





- Trained with movie reviews



# 02. 3주차 강의 | 워드 임베딩

## 과제 안내

22-2 KUBIG NLP 분반 > Week3 > 과제 샘플 ▾

이름 ↑	마지막으로 수정한 날짜	파일 크기
 negative sampling_미드 '프렌즈'	오후 4:37 나	84KB
 예브리타임 게시물 word2vec & t-SNE 시각화.ipynb	오후 4:36 나	235KB
 eng_kor_word2vec_네이버영화리뷰.ipynb	오후 4:20 나	36KB
 negative_sampling_뉴스.ipynb	오후 4:09 나	81KB

### 다양한 과제 옵션

(아래는 단순히 어떤 과제를 구현해볼지에 대한 옵션을 드릴뿐이지, 꼭 아래 옵션에서 고르시지 않으셔도 됩니다 !)

- 1. 제공된 데이터셋 혹은 본인이 직접 선택한 데이터셋을 통해서 간단한 CBOW/Skip Gram 모델 훈련해보기
- 2. t-sne를 활용해서 직접 훈련한 Word2Vec 모델에 다양한 단어를 입력해보며 시각화해보기
- 3. 직접 훈련한 Word2Vec 모델에 관심있는 단어를 입력하고 해당 단어와 유사한 단어들 살펴보기
- 4. 특정 두 단어 사이의 유사도(거리)를 측정해보고 모델이 잘 훈련되었는지 판단해보기
- 5. Word2Vec 모델 훈련 시 window\_size 등 다양한 초모수 설정 변경해가며 결과 비교해보기

