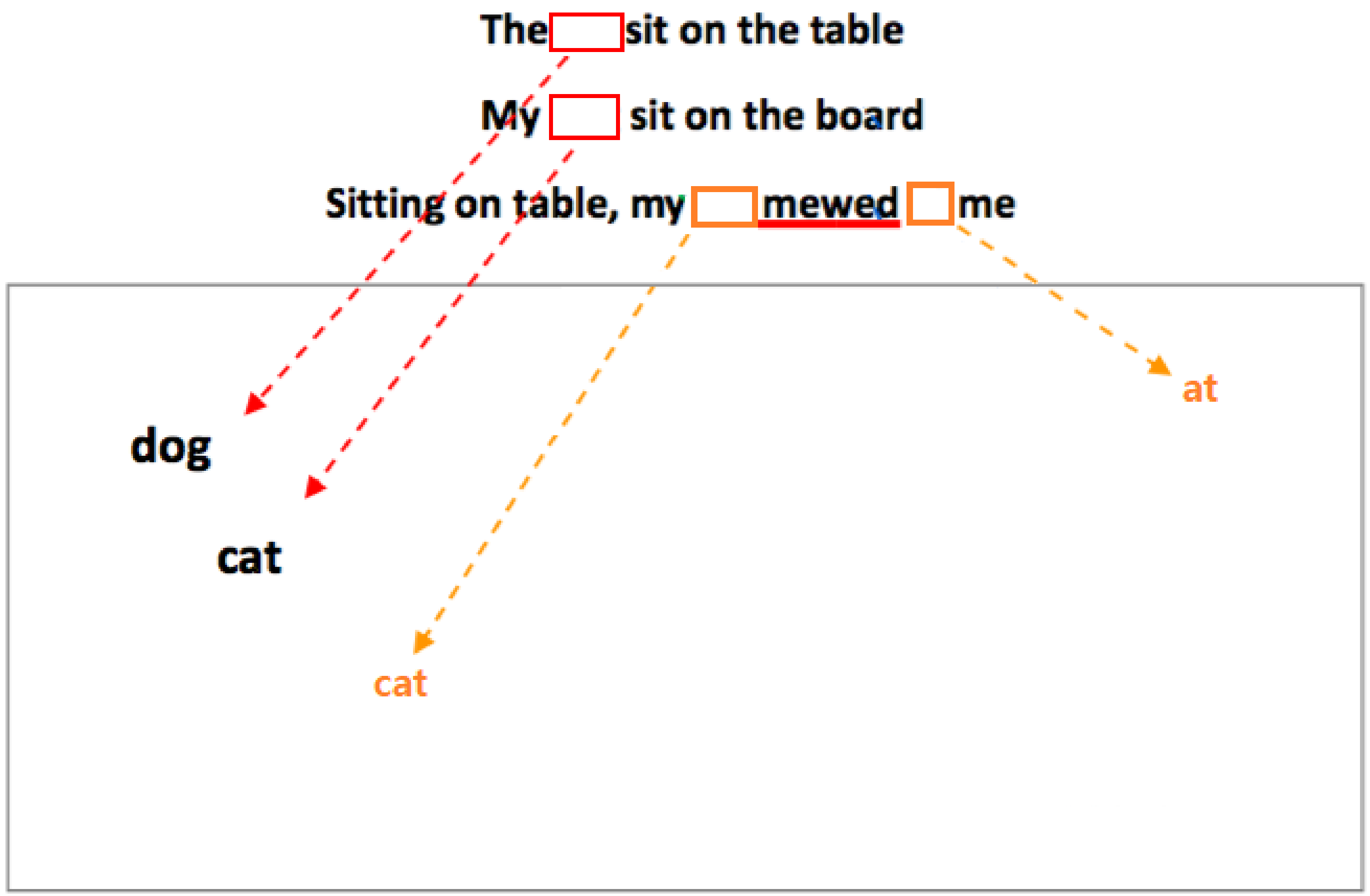
WORK2VEC

JuHyeong Kim

WORD2VEC

- The basic idea is **that either a word is used to predict the context of it** or **the other way around-to use the context to predict a current word.**

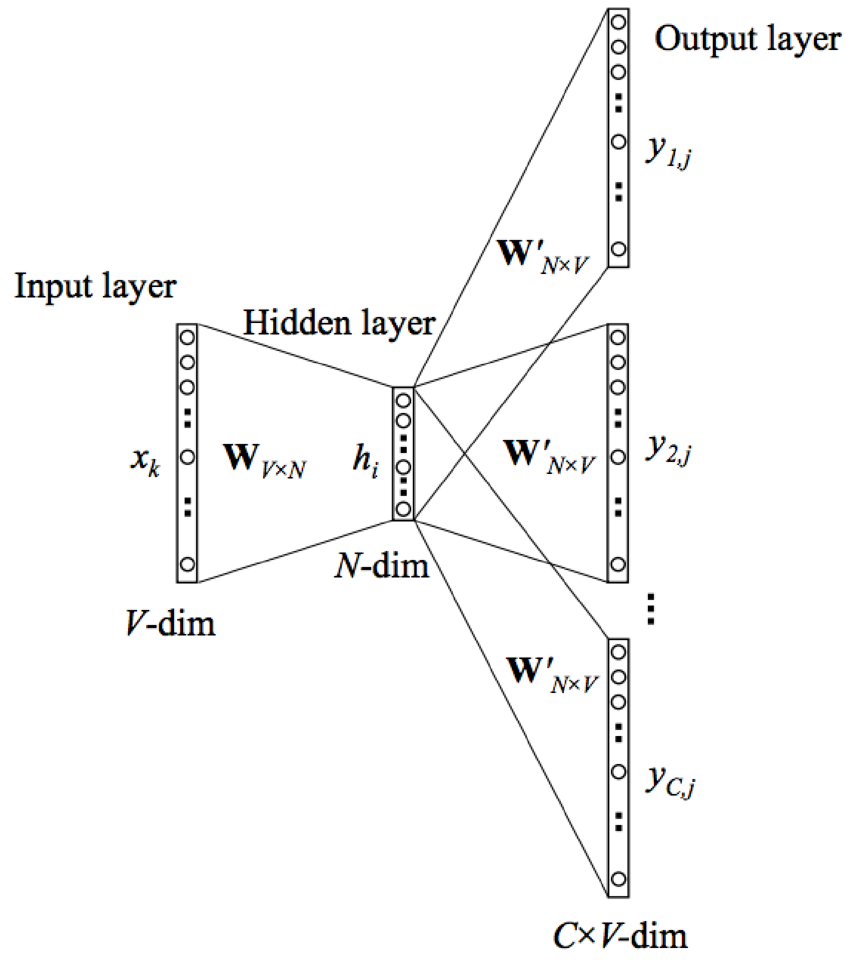


WORD2VEC

- Input : word (ONE-HOT Vector)
- output : word (Distributed Representation)

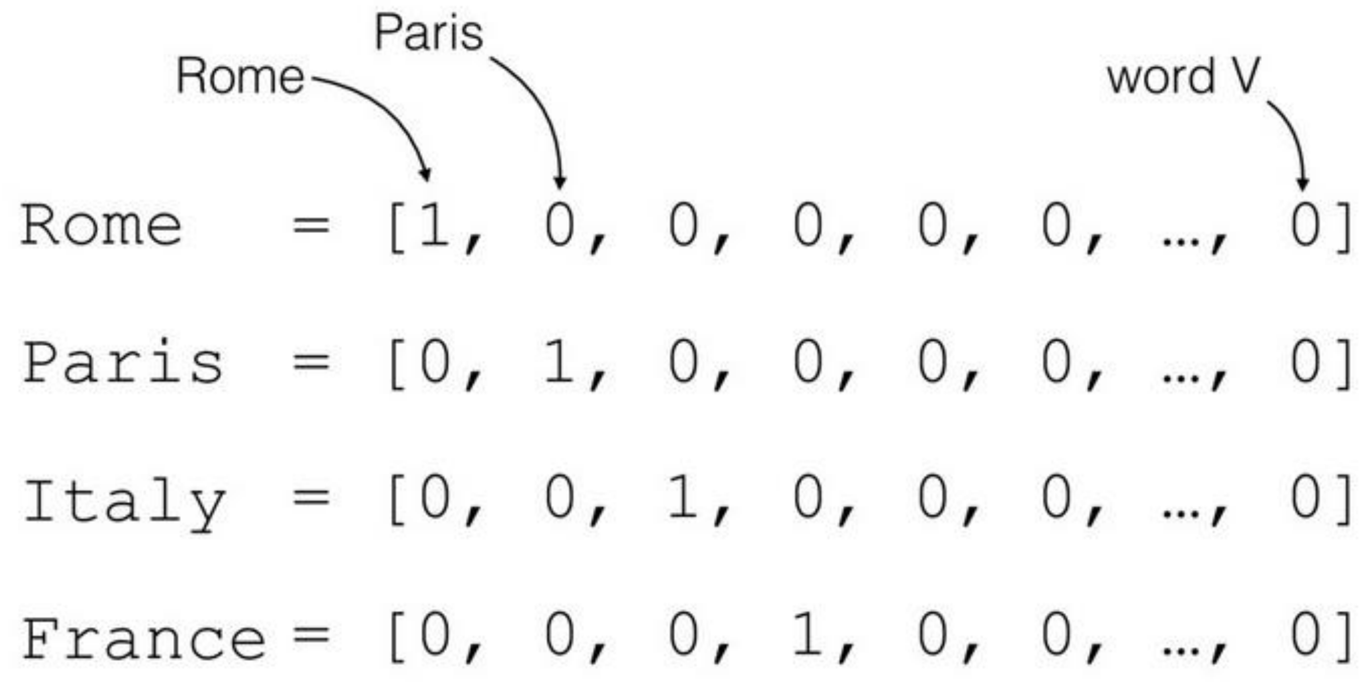
- window size : 2
- Hidden layer : 1

- loss function : cross-entropy



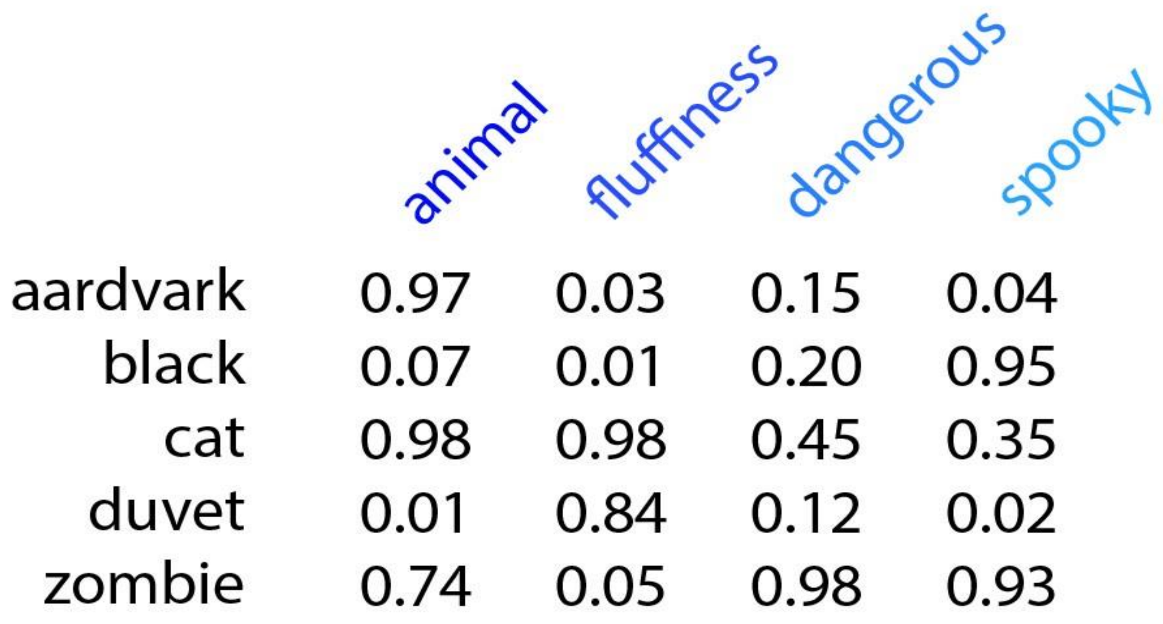
ONE-HOT VECTOR

- A one hot encoding is a representation of categorical variables as binary vectors.
- Each integer value is represented as a binary vector that is **all zero values except the index of the integer**, **which is marked with a 1**.


Rome
Paris
word V
Rome = [1, 0, 0, 0, 0, 0, ..., 0]
Paris = [0, 1, 0, 0, 0, 0, ..., 0]
Italy = [0, 0, 1, 0, 0, 0, ..., 0]
France = [0, 0, 0, 1, 0, 0, ..., 0]

DISTRIBUTED REPRESENTATION

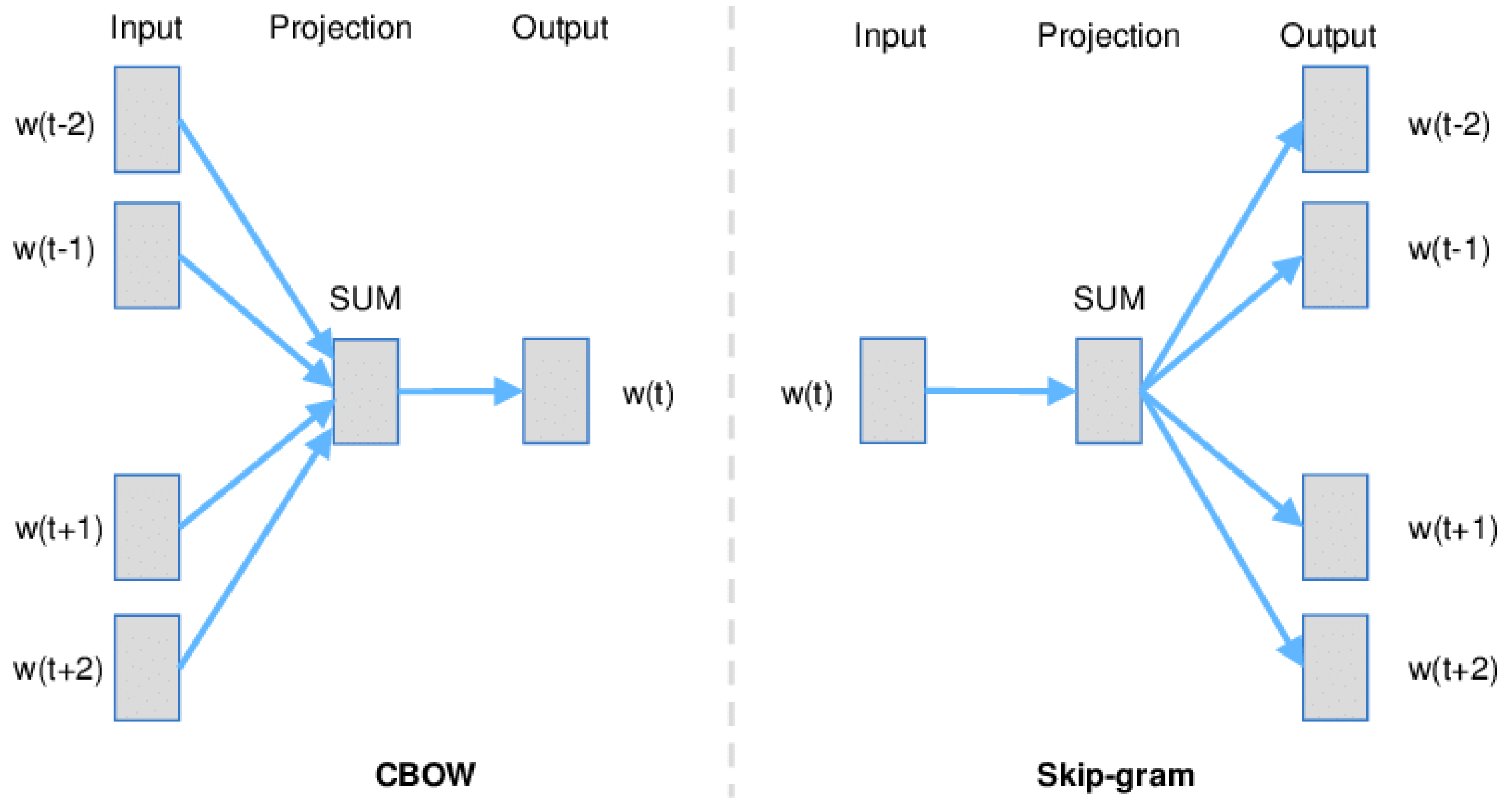
- as humans speaking some language, we know that words are these rich entities with many layers of connotation and meaning.
- Let’s hand-craft some semantic features for these 5 words.
- Specifically, **let’s represent each word as having some sort of value between 0 and 1 for four semantic qualities**, “animal”, “fluffiness”, “dangerous”, and “spooky”:



	animal	fluffiness	dangerous	spooky
aardvark	0.97	0.03	0.15	0.04
black	0.07	0.01	0.20	0.95
cat	0.98	0.98	0.45	0.35
duvet	0.01	0.84	0.12	0.02
zombie	0.74	0.05	0.98	0.93

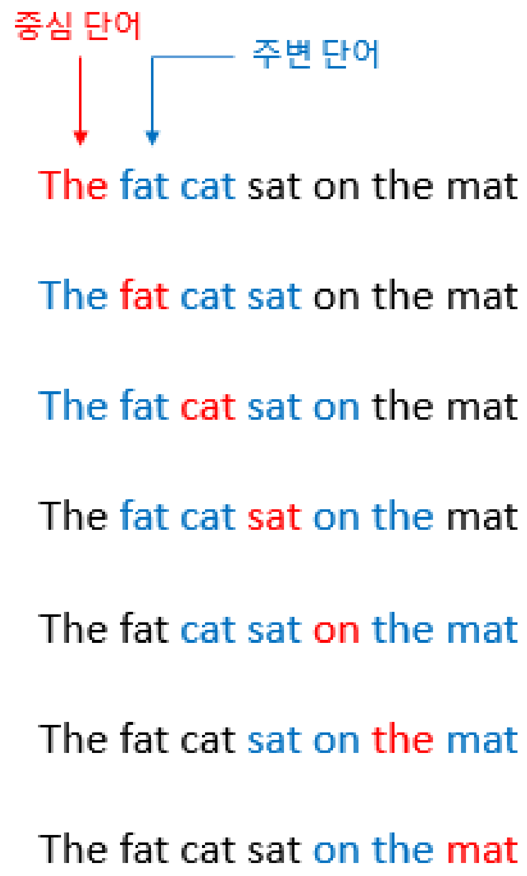
WORD2VEC

- Word2Vec has two methods: Continuous Bag of Words (CBOW) and Skip-Gram.



CONTINUOUS BAG OF WORDS (CBOW)

- window size : 2



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

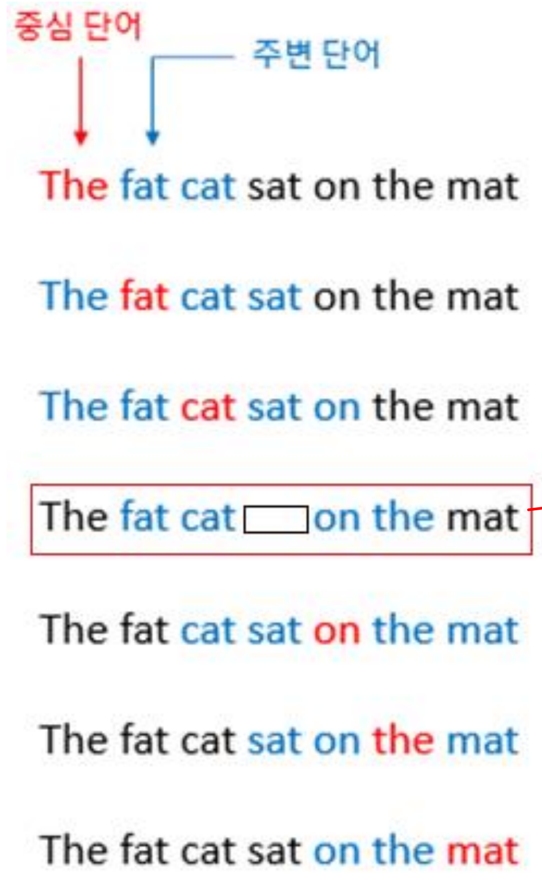
The fat cat sat on the mat

The fat cat sat on the mat

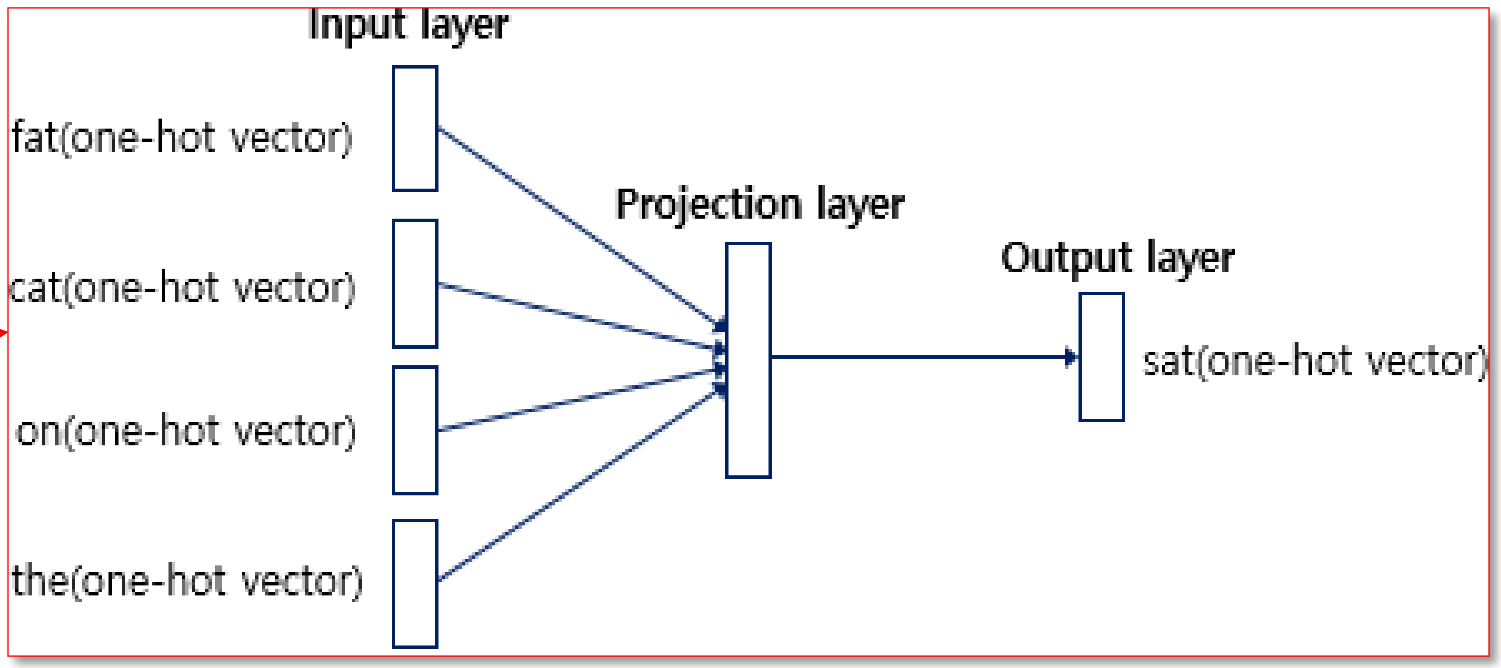
The fat cat sat on the mat

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]

CONTINUOUS BAG OF WORDS (CBOW)

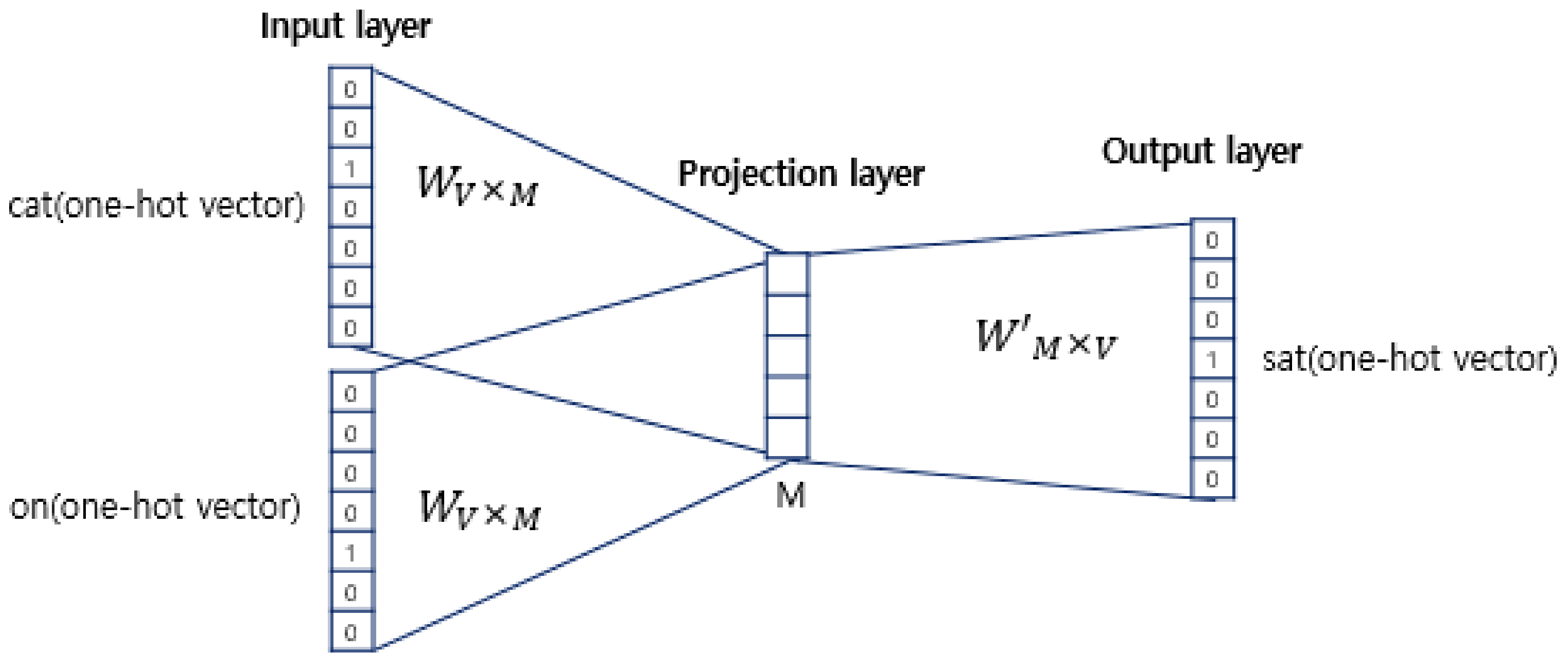


중심 단어
주변 단어
The fat cat sat on the mat
The fat cat sat on the mat
The fat cat sat on the mat
The fat cat on the mat
The fat cat sat on the mat
The fat cat sat on the mat
The fat cat sat on the mat

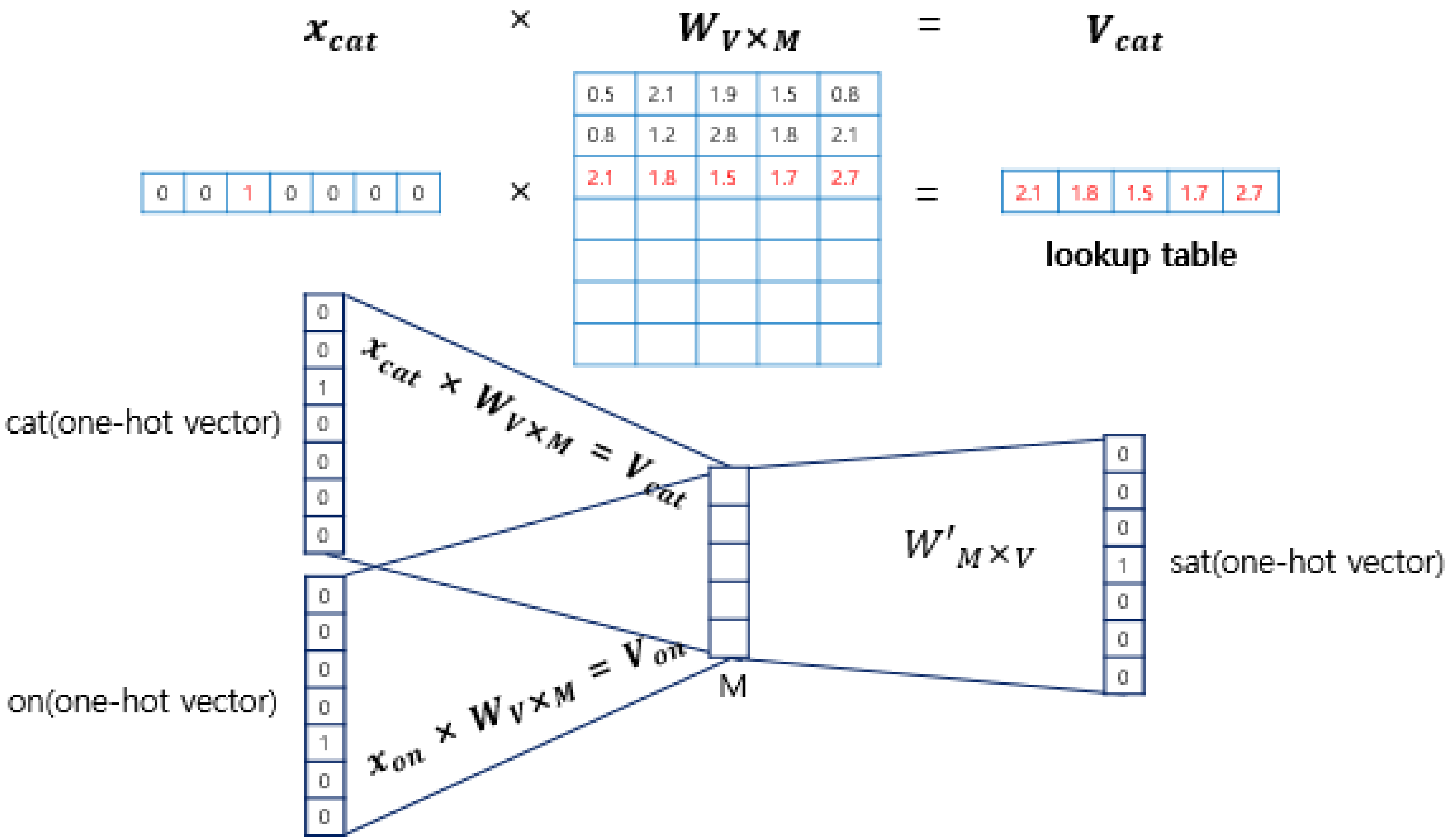


CONTINUOUS BAG OF WORDS (CBOW)

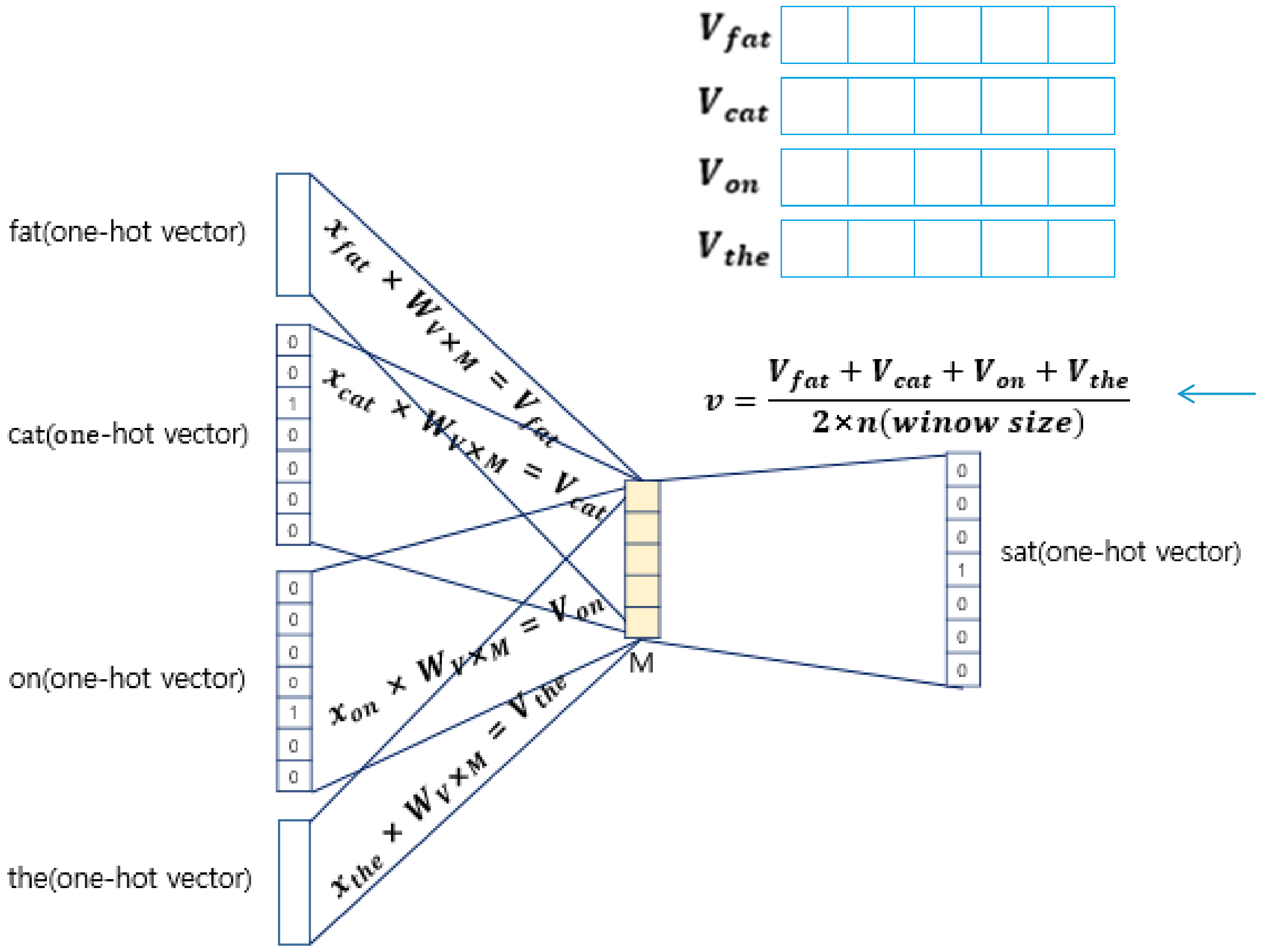
- The value of M is set **arbitrarily.**



CONTINUOUS BAG OF WORDS (CBOW)



CONTINUOUS BAG OF WORDS (CBOW)

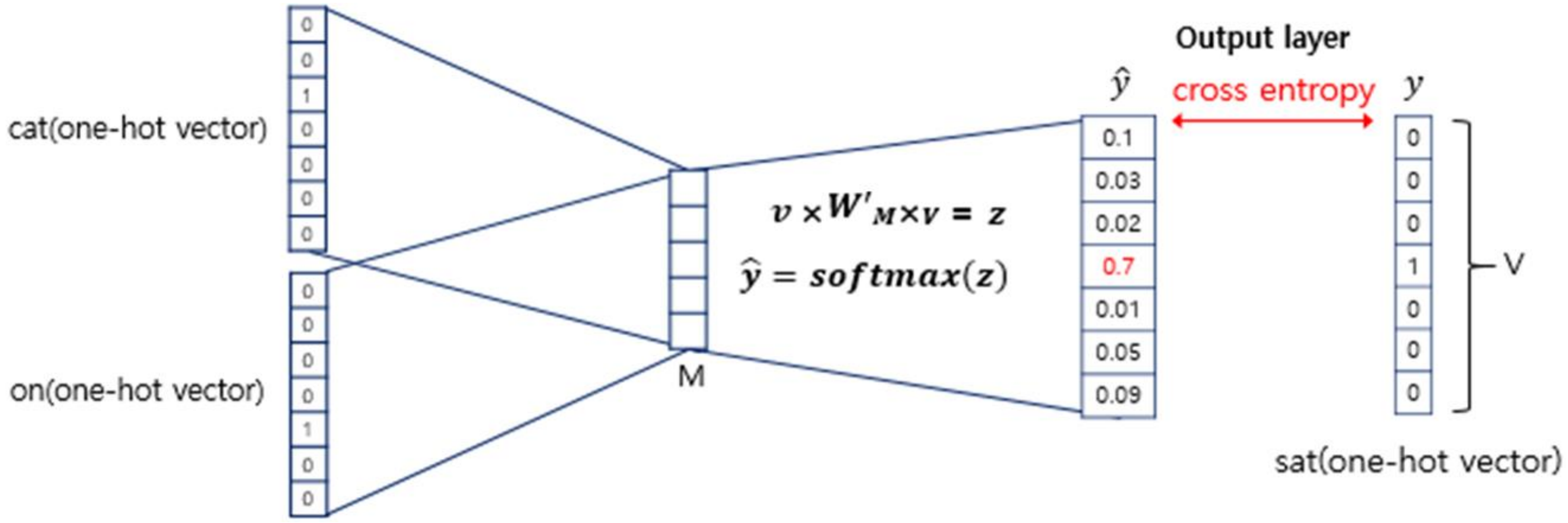


V_{fat}
V_{cat}
V_{on}
V_{the}
fat(one-hot vector)
cat(one-hot vector)
on(one-hot vector)
the(one-hot vector)
sat(one-hot vector)
$x_{fat} \times W_{V \times M} = V_{fat}$
$x_{cat} \times W_{V \times M} = V_{cat}$
$x_{on} \times W_{V \times M} = V_{on}$
$x_{the} \times W_{V \times M} = V_{the}$
M

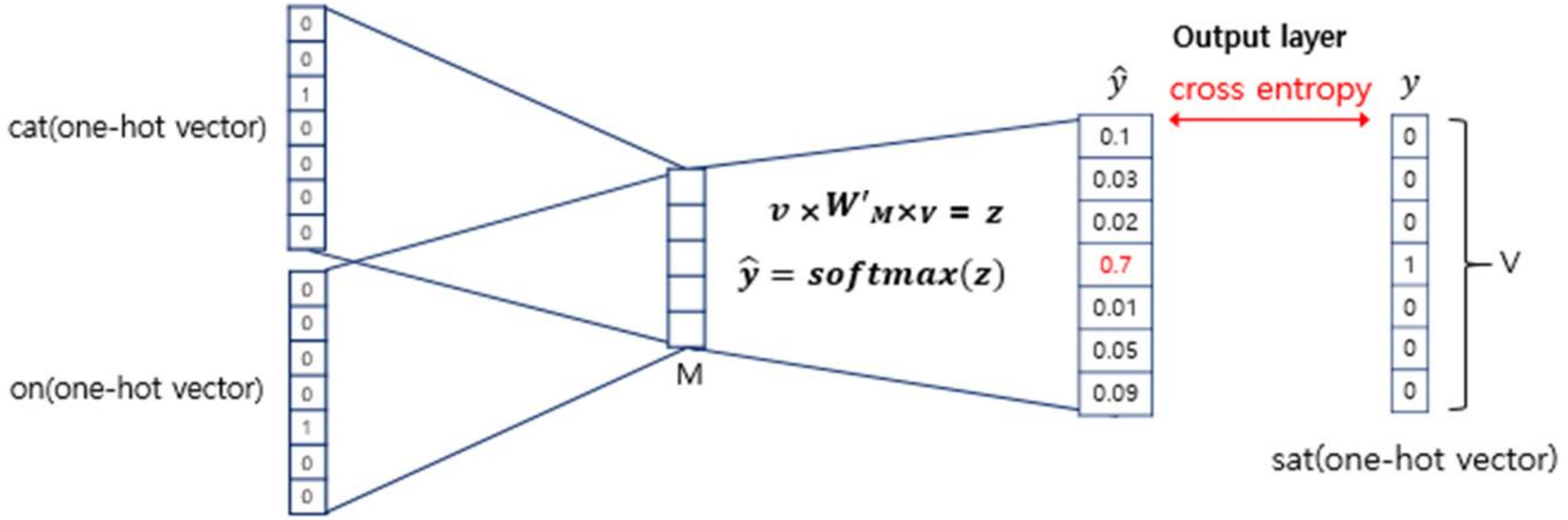
$$v = \frac{V_{fat} + V_{cat} + V_{on} + V_{the}}{2 \times n(winow\ size)}$$

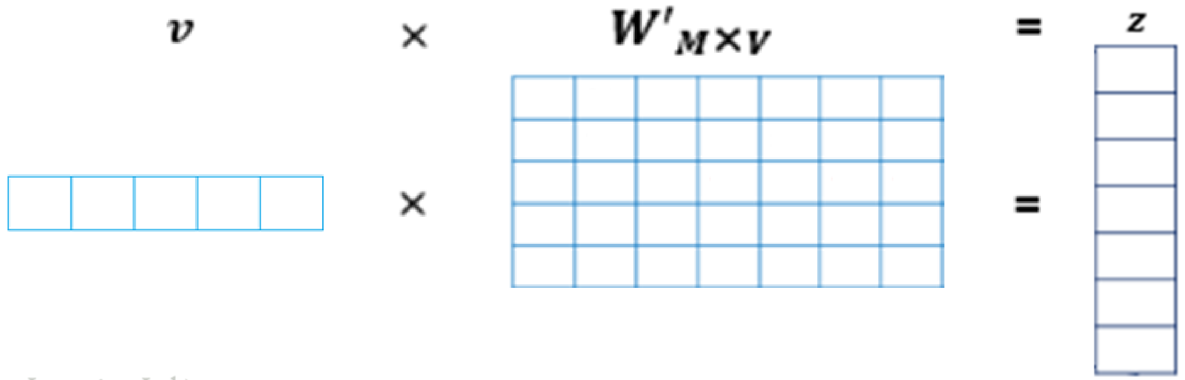
Find the average.

CONTINUOUS BAG OF WORDS (CBOW)



CONTINUOUS BAG OF WORDS (CBOW)

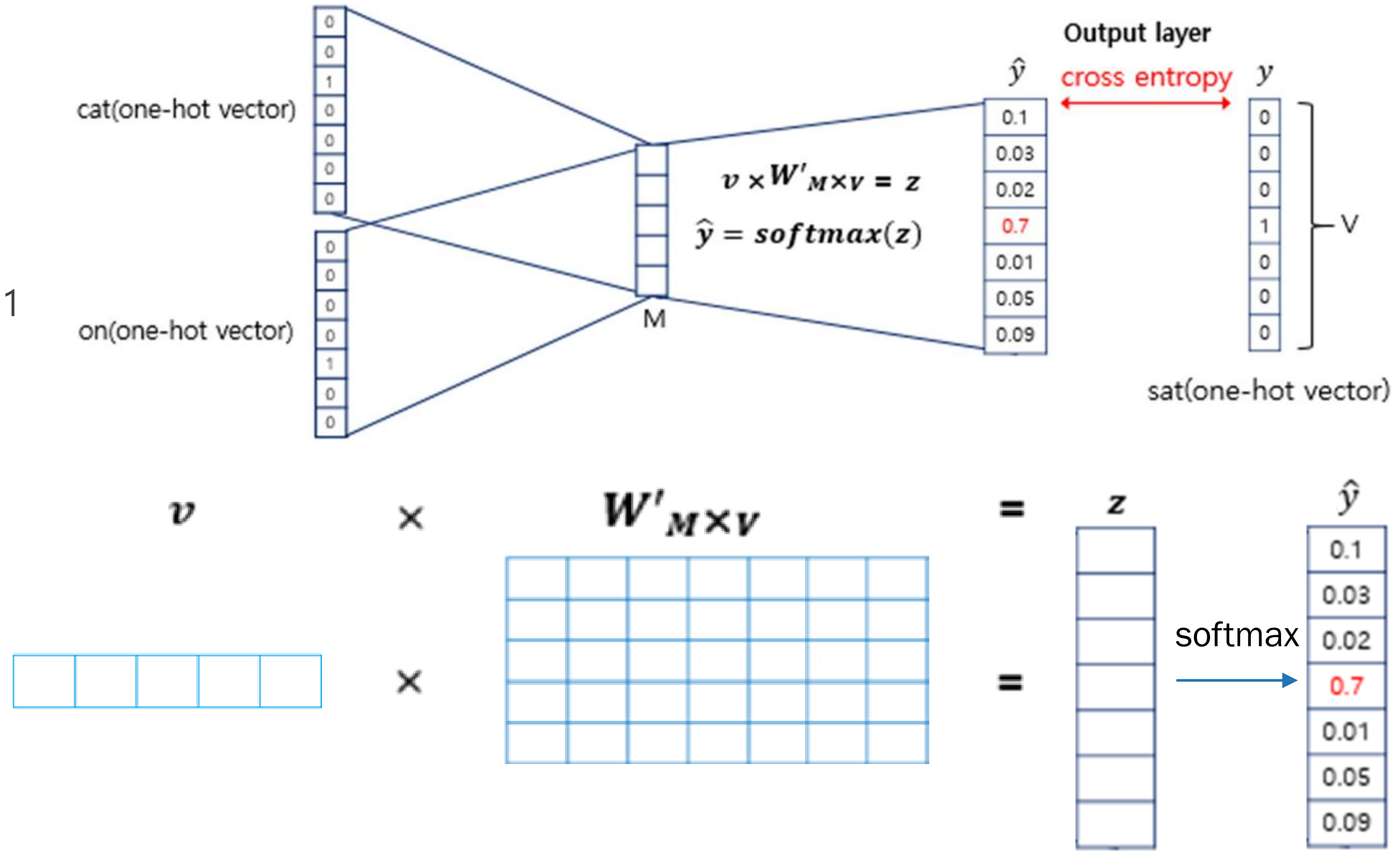




CONTINUOUS BAG OF WORDS (CBOW)

- Softmax
  - Probability values of all classes are obtained by normalizing the results of multiple operations
  - Convert the input to a value between 0 and 1
  - Make the total for the converted result equal to 1

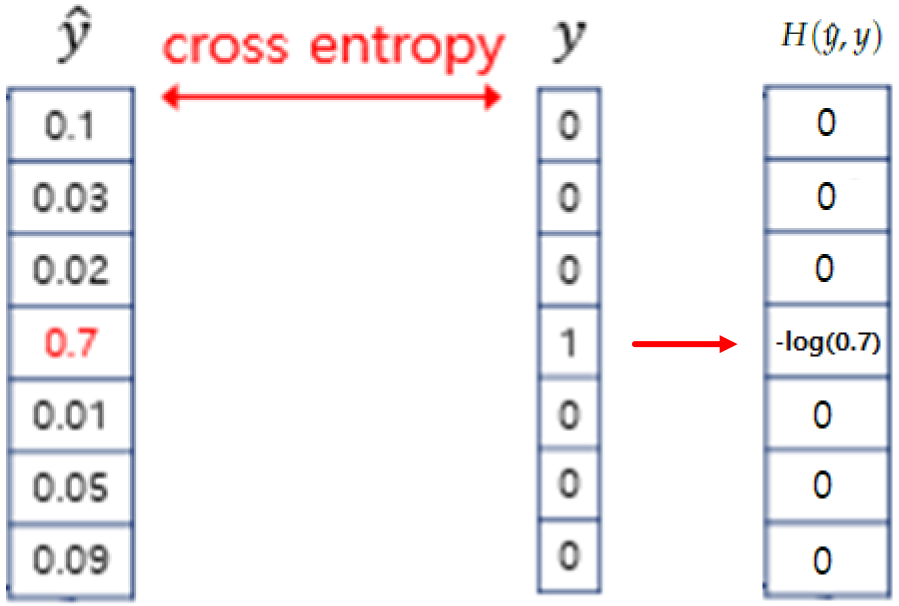
$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$



CONTINUOUS BAG OF WORDS (CBOW)

- Weight learning
  - Learning by minimizing loss function using Backpropagation
- cross entropy

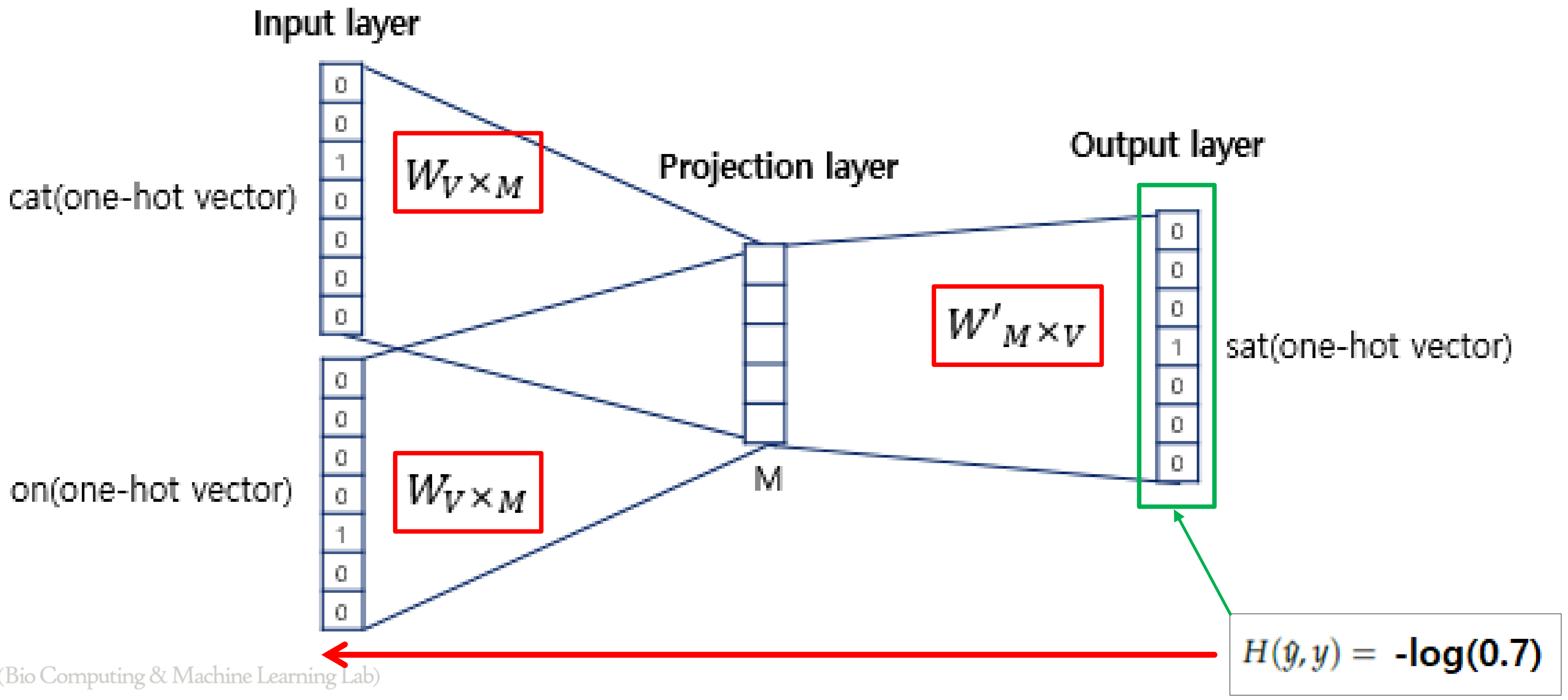
$$H(\hat{y}, y) = -\sum_{j=1}^{|V|} y_j \log(\hat{y}_j)$$



$$H(\hat{y}, y) = -\log(0.7)$$

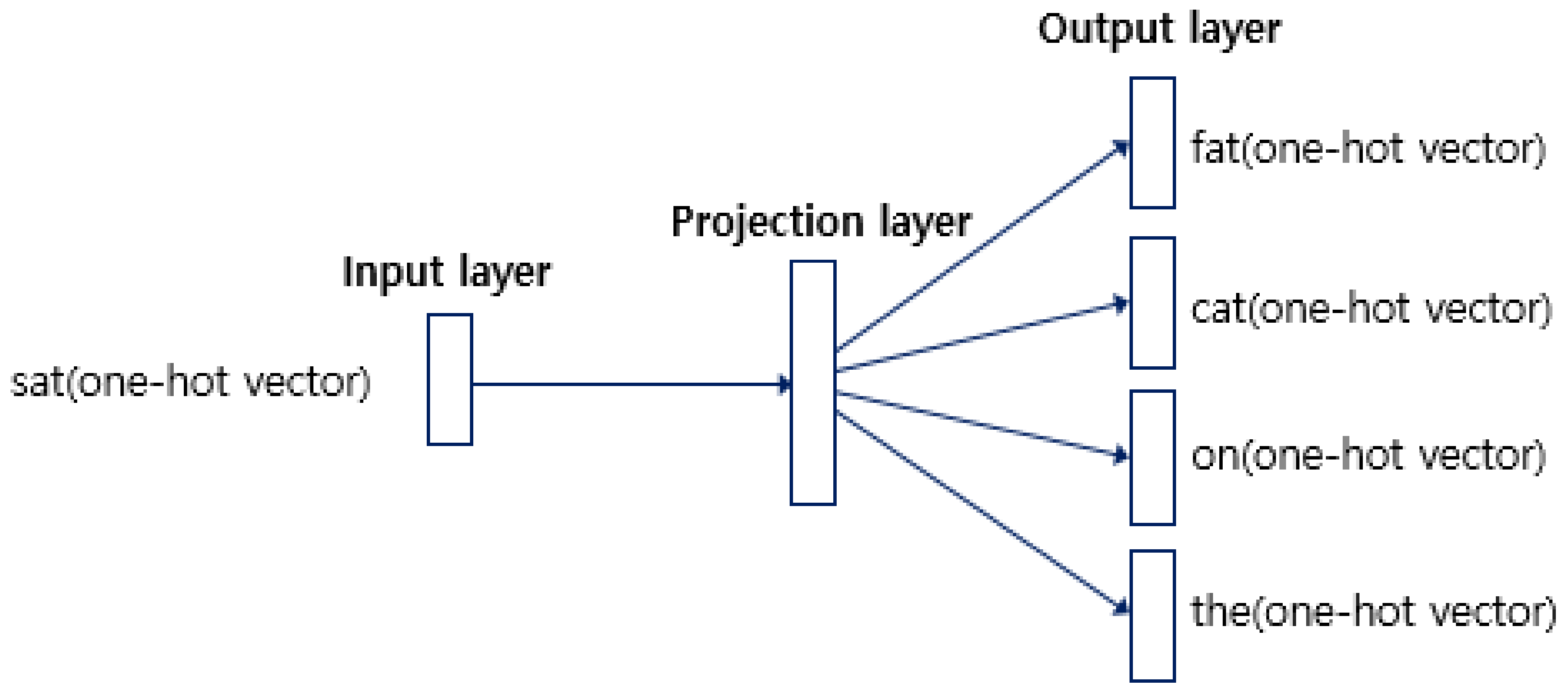
CONTINUOUS BAG OF WORDS (CBOW)

- Backpropagation
  - Update the weight value using the loss function.



SKIP-GRAM

- Skip-gram predicts surrounding words from the central word.



REFERENCE

- https://wikidocs.net/22660
- http://www.claudiobellei.com/2018/01/06/backprop-word2vec/
- https://operatingsystems.tistory.com/entry/Data-Mining-Word2vec-CBOW
- https://ronxin.github.io/wevi/