23.07.20 _ 6주차

딥러닝 논문 요약 및 구현 스터디

발표자

정명찬

Word 2 Vector

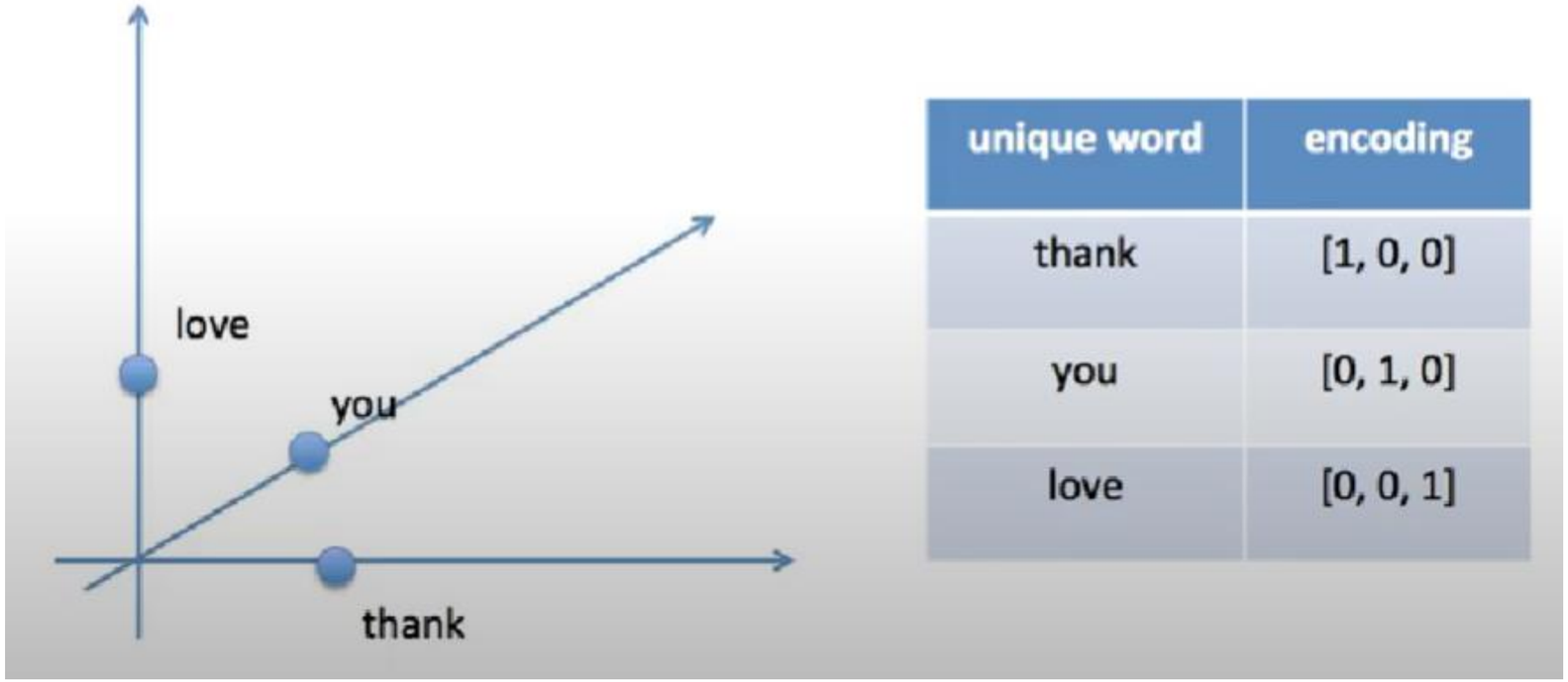
Efficient Estimation of Word Representations in Vector Space

Abstract

- 단어를 학습하기 위해 숫자로 변환 필요 -> 원-핫 인코딩
- 원-핫 인코딩의 한계를 극복하기 위해 워드 임베딩 필요
- 기존의 워드 임베딩(NNLM, RNN)의 한계를 극복하기 위해 word2vec 제안

- 단어의 벡터 표현을 계산하기 위한 새로운 구조
- 구문적, 의미적 단어 유사성을 측정하는 데에 높은 성능

I. Introduction



unique word	encoding
thank	[1, 0, 0]
you	[0, 1, 0]
love	[0, 0, 1]

Encoding vs Embedding

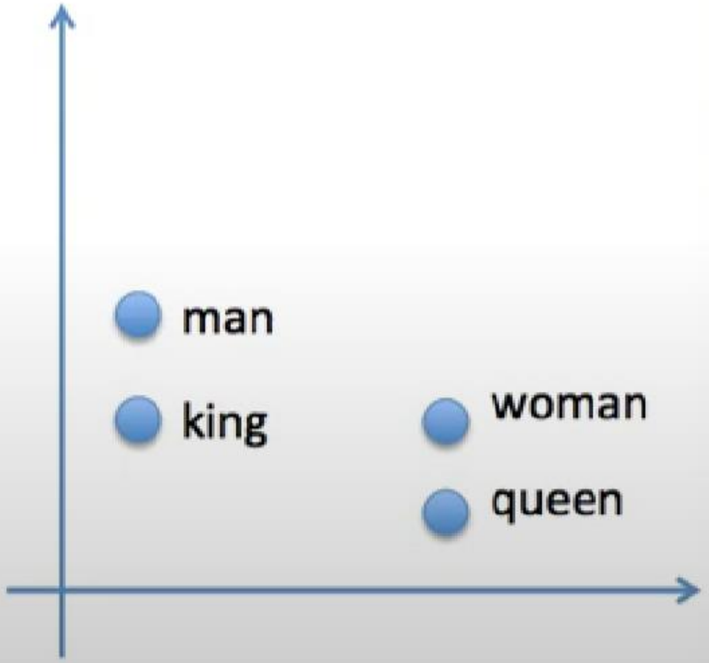
원-핫 인코딩만으로는 단어의 의미를 포함할 수 없음

단어 간의 유사성 측정 불가

단어의 개수가 많아지면 벡터 차원도 비례해서 늘어남

I. Introduction

unique word	encoding	embedding
king	[1, 0, 0, 0]	[1, 2]
man	[0, 1, 0, 0]	[1, 3]
queen	[0, 0, 1, 0]	[5, 1]
woman	[0, 0, 0, 0]	[5, 3]



Encoding vs **Embedding**

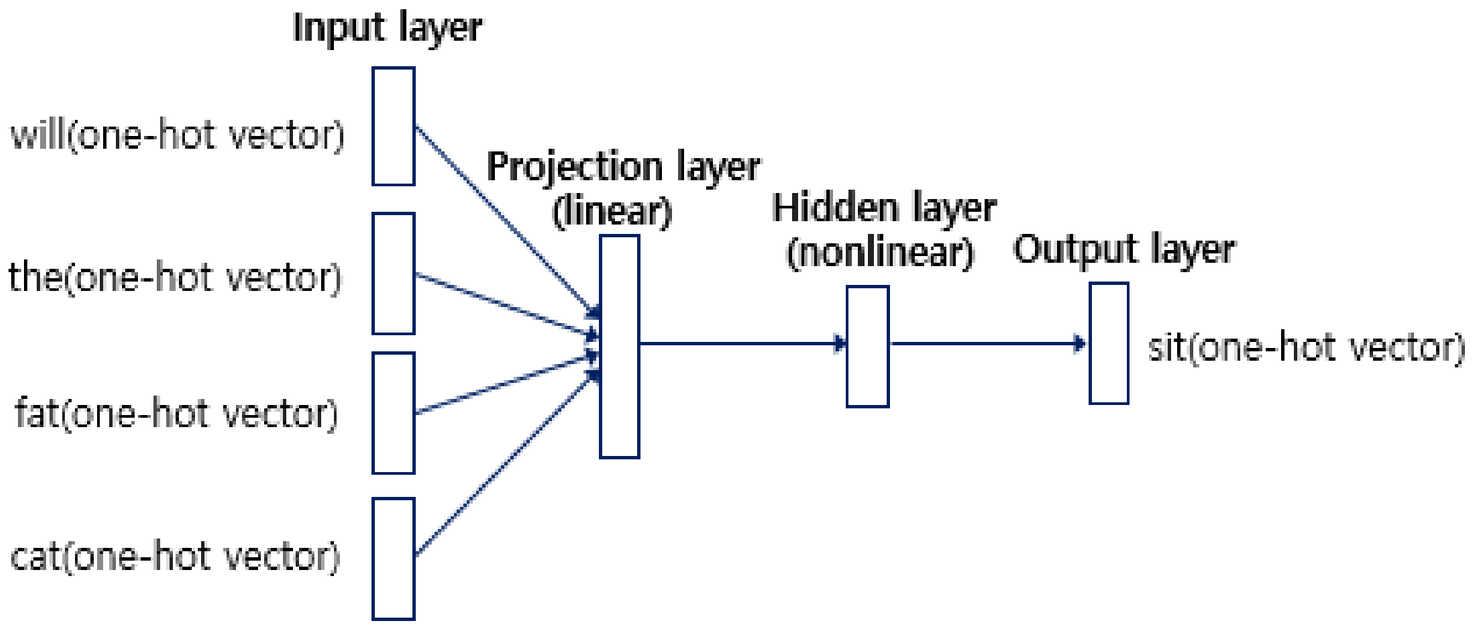
임베딩을 이용하면 비슷한 의미를 지닌 단어끼리 가깝게 위치하도록 표현 가능

단어 벡터의 차원을 임의로 조절(축소) 가능

II. Model Architecture

VS **NNLM(Neural Net Language Model)**

주어진 문장: What will the fat cat **sit** on? (window: 4, 'sit' 예측)



II. Model Architecture

VS **NNLM(Neural Net Language Model)**

주어진 문장: What will the fat cat **sit** on? (window: 4, 'sit' 예측)

x_{fat} × $W_{V \times M}$ = e_{fat}

0	0	0	1	0	0	0

×

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

=

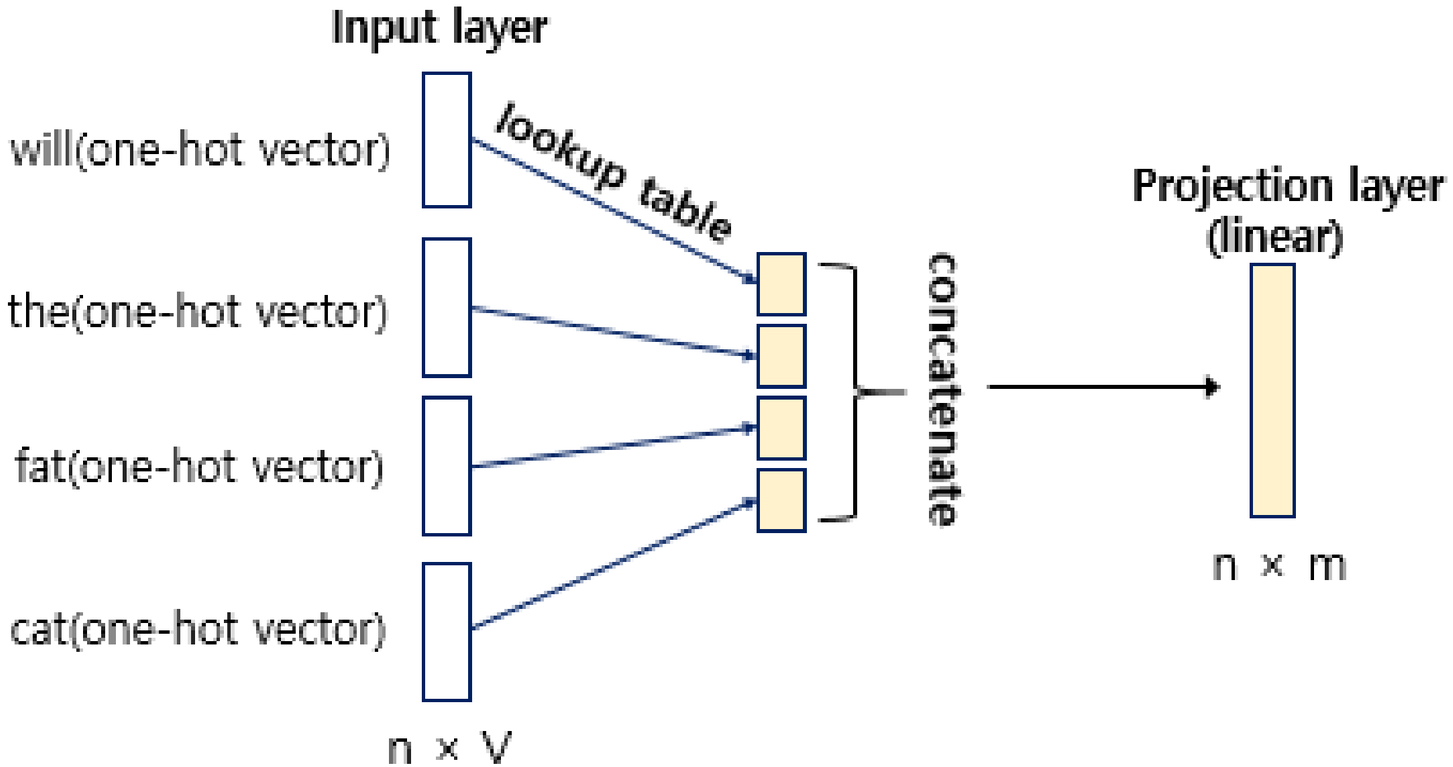
2.1	1.8	1.5	1.7	2.7

lookup table

II. Model Architecture

VS **NNLM(Neural Net Language Model)**

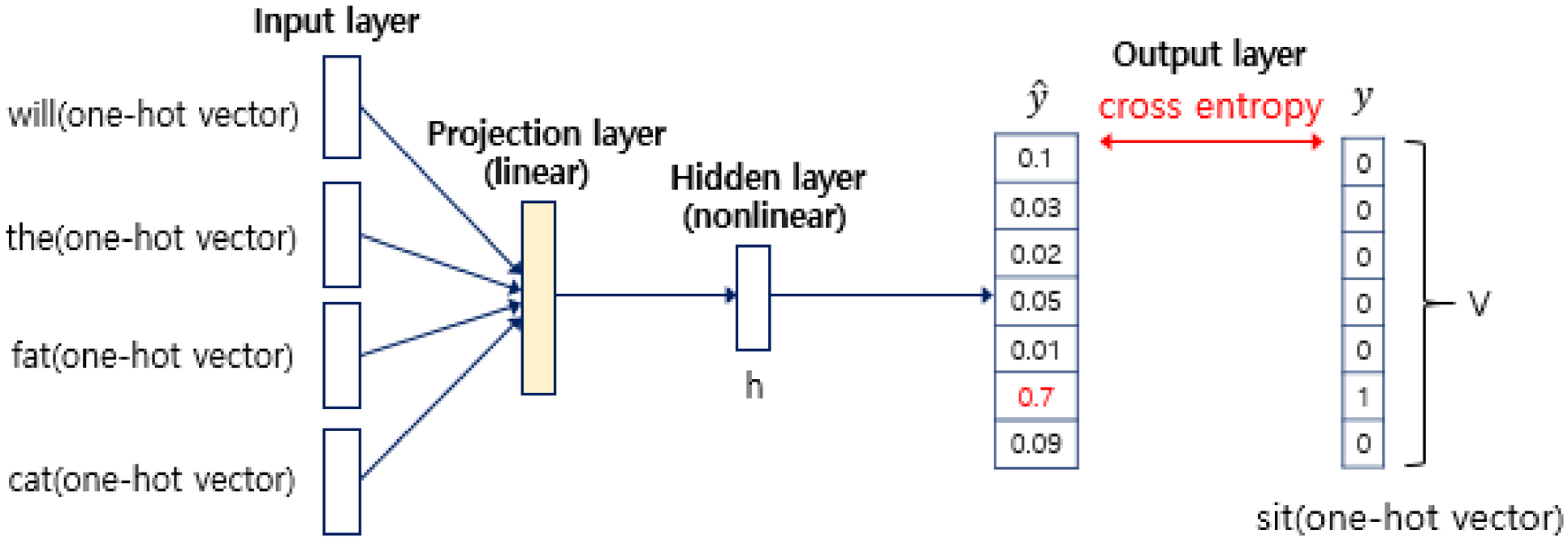
주어진 문장: What will the fat cat **sit** on? (window: 4, 'sit' 예측)



II. Model Architecture

VS **NNLM(Neural Net Language Model)**

주어진 문장: What will the fat cat **sit** on? (window: 4, 'sit' 예측)

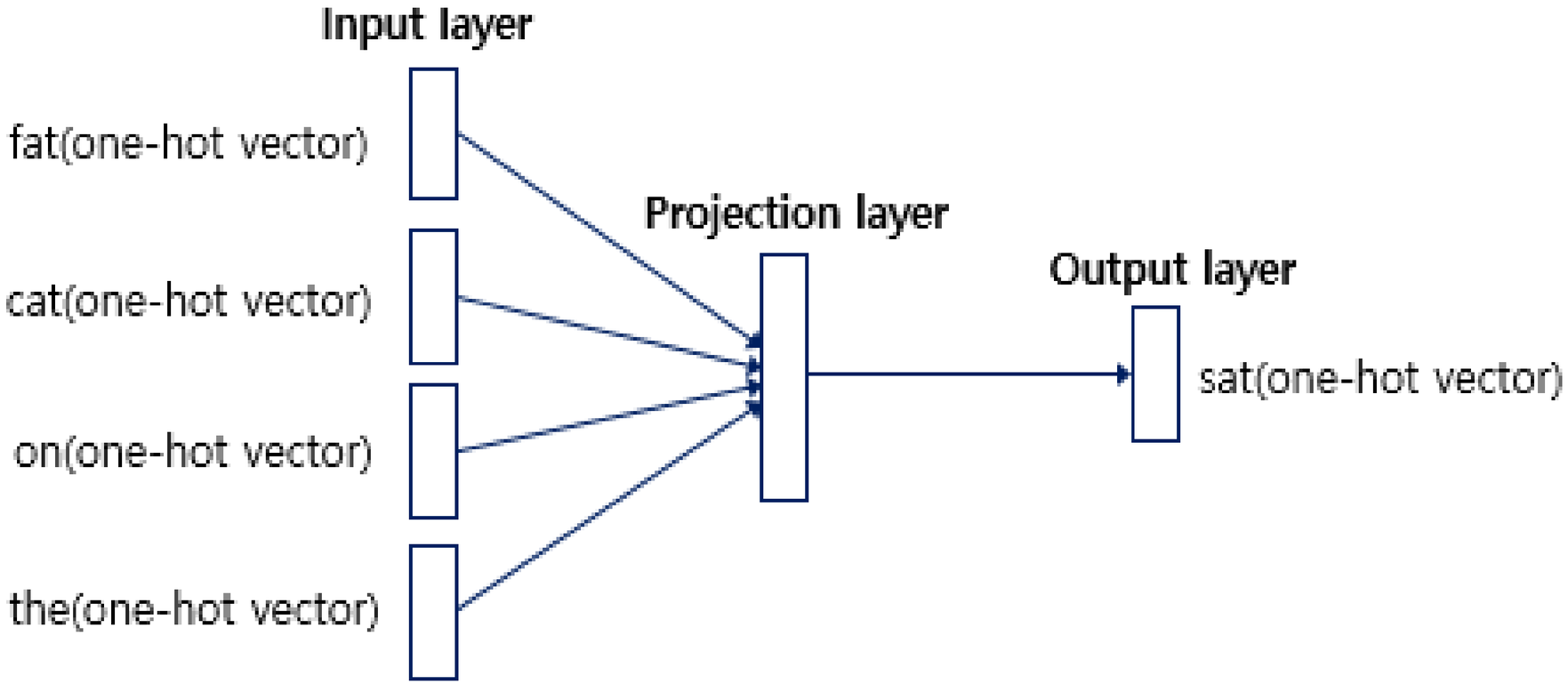


III. New Log-linear Models

Word2Vec – CBOW(continuous bag of words)

The fat cat **sat** on the table. (window: 2, 'sat' 예측)

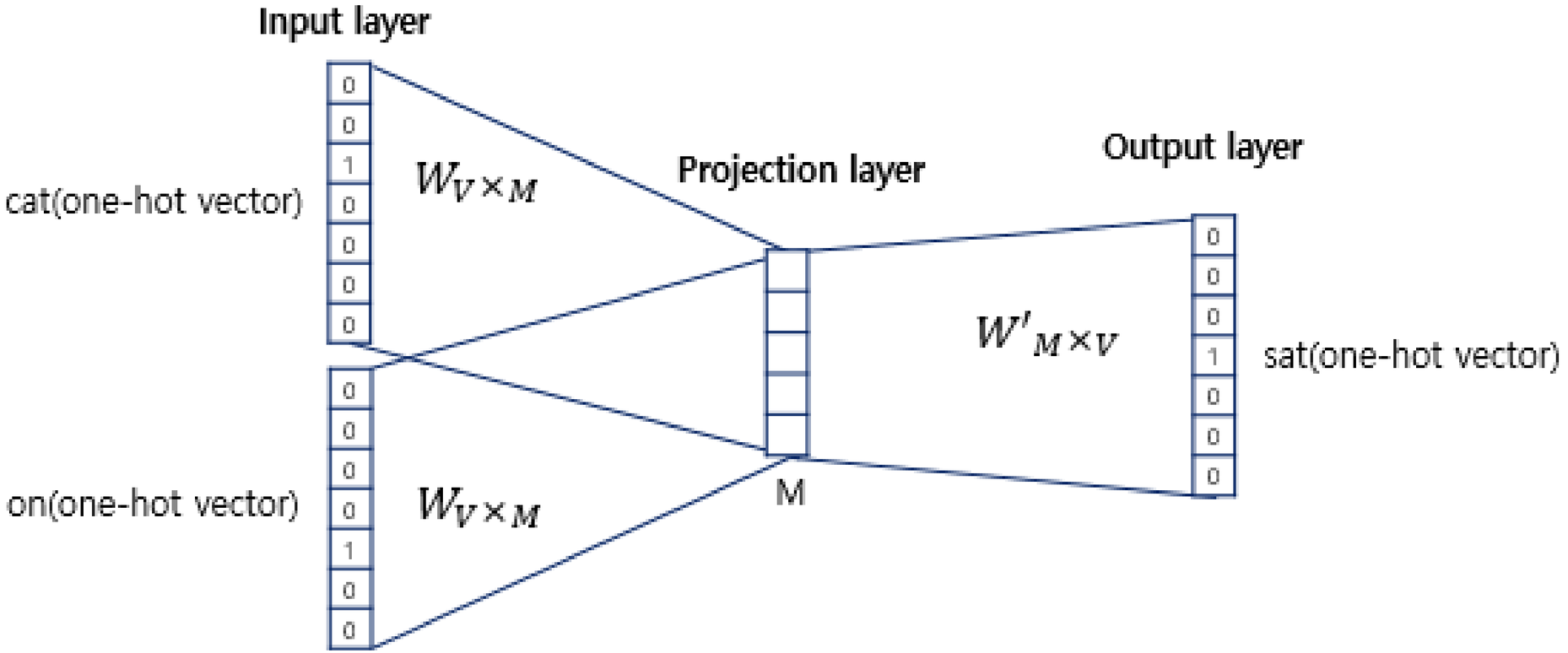
다음에 나올 단어가 아닌 주변 좌2, 우2 단어를 가지고 기준 단어 예측



III. New Log-linear Models

Word2Vec – CBOW(continuous bag of words)

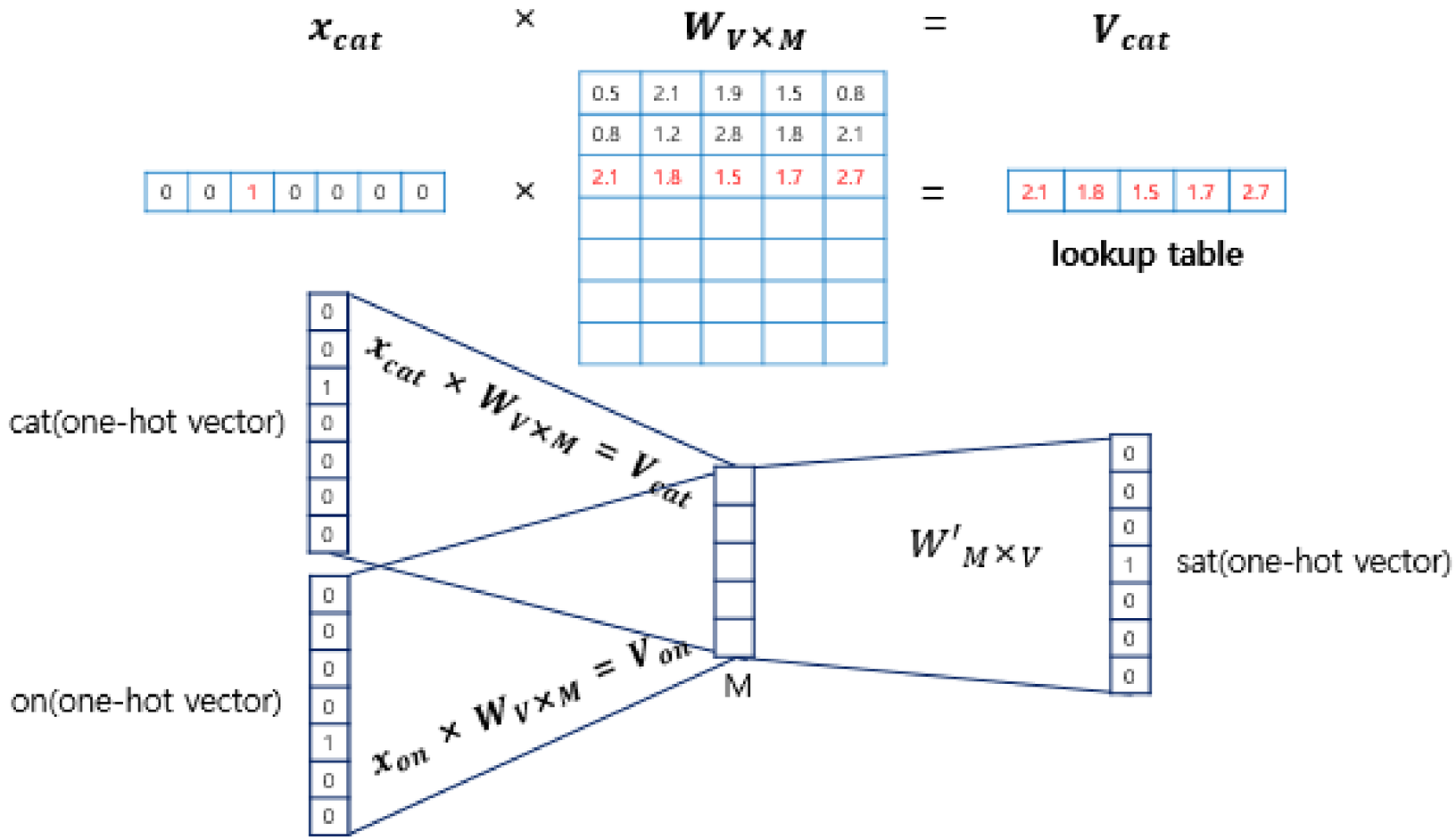
W, W′ 별개의 두 가중치가 학습이 되고, sat의 임베딩 벡터가 projection layer의 M차원 벡터



III. New Log-linear Models

Word2Vec – CBOW(continuous bag of words)

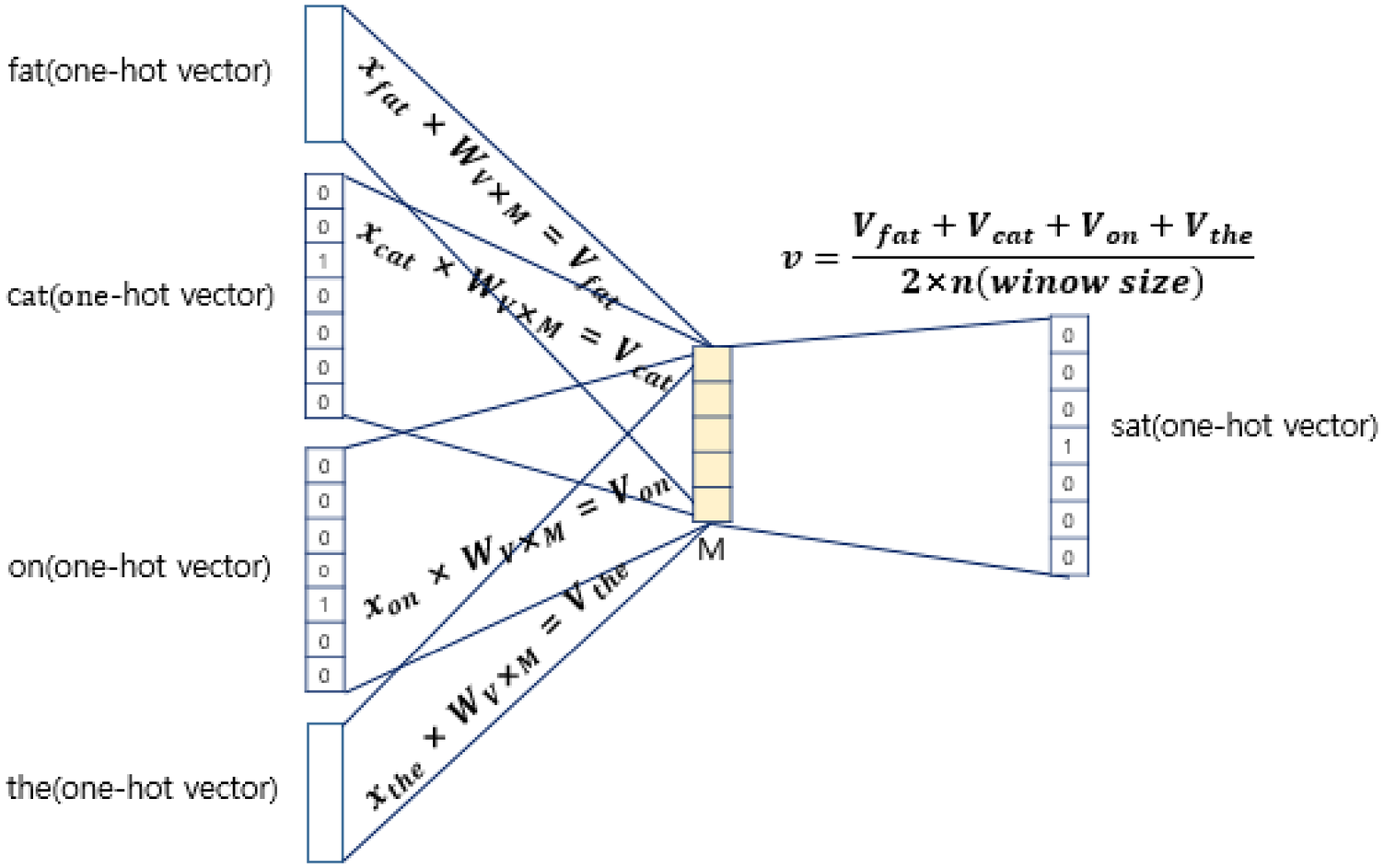
NNLM과 달리, concat 하지 않고, 평균을 구해서 1xM을 만든다.



III. New Log-linear Models

Word2Vec – CBOW(continuous bag of words)

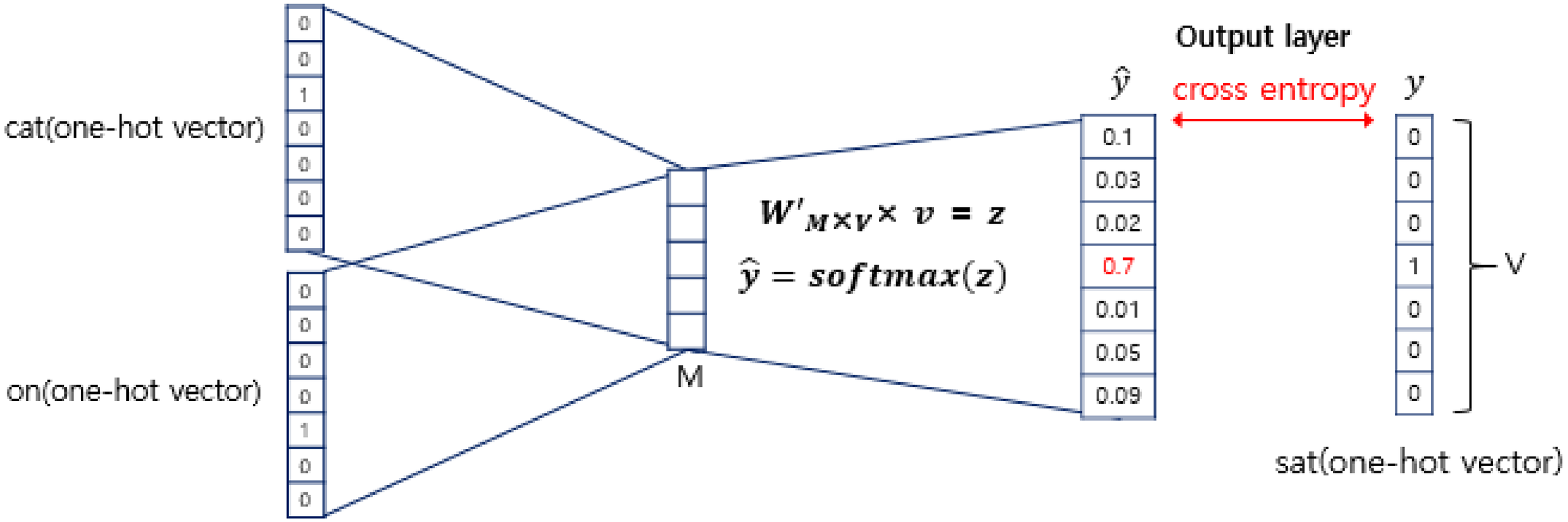
NNLM과 달리, concat 하지 않고, 평균을 구해서 1xM을 만든다.



III. New Log-linear Models

Word2Vec – CBOW(continuous bag of words)

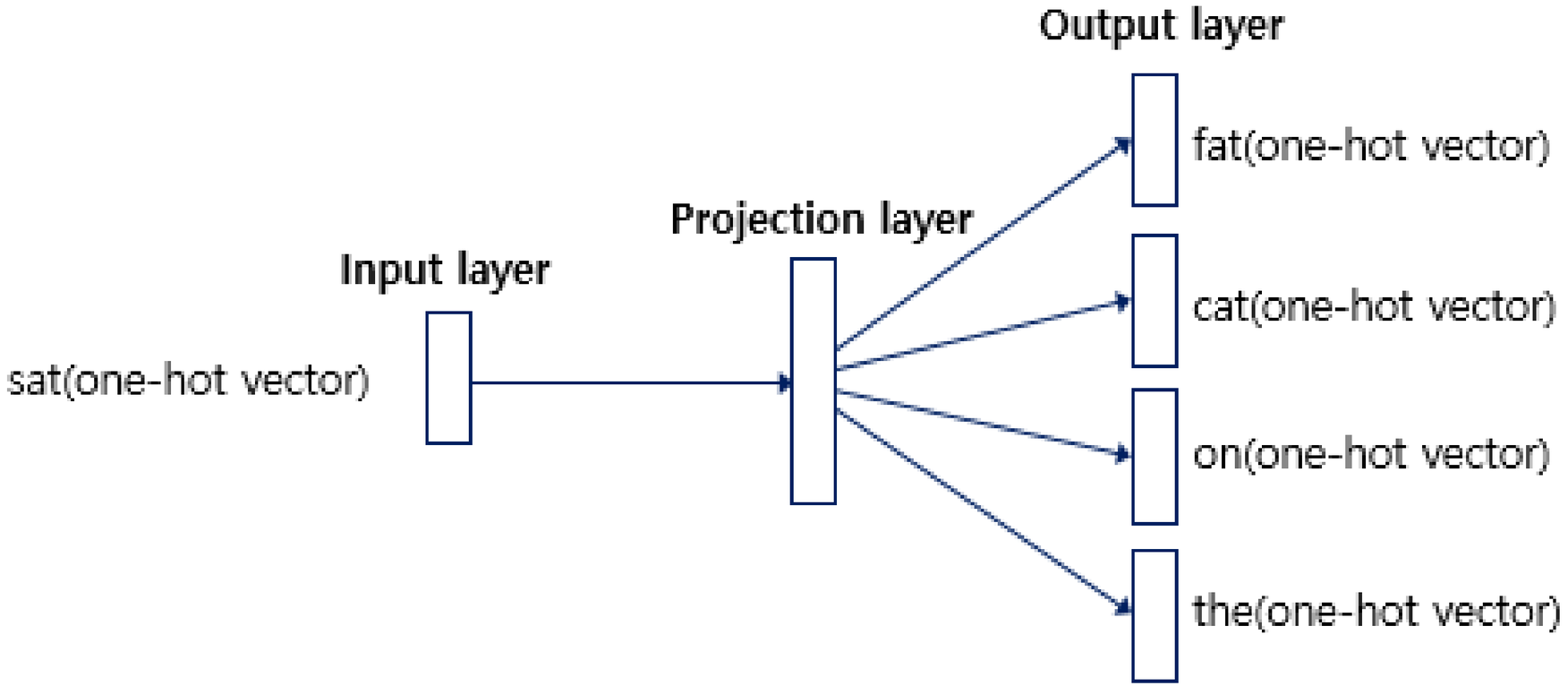
NNLM과 달리, 중간에 Non-linear hidden layer가 없음



III. New Log-linear Models

Word2Vec – skip-gram

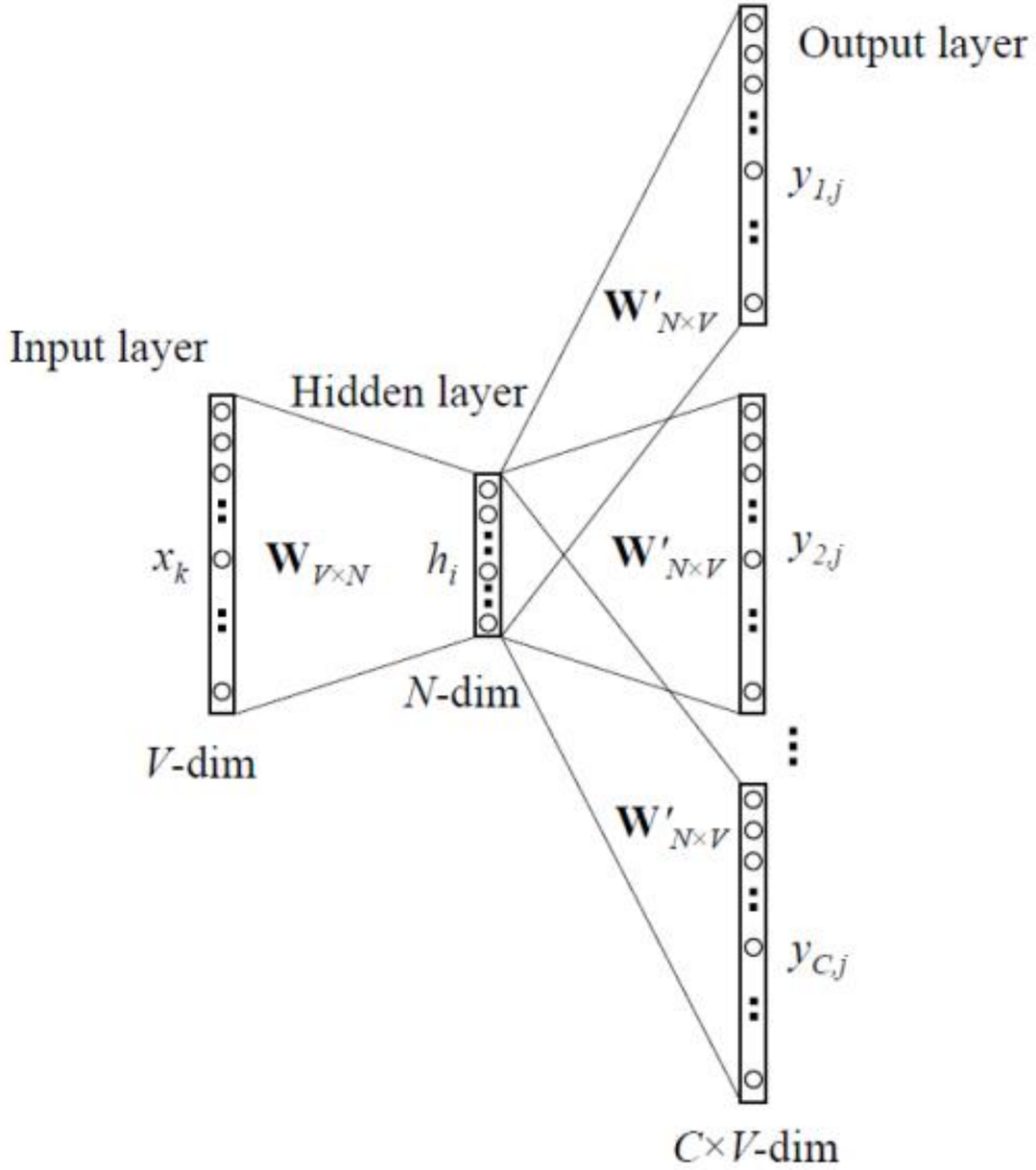
‘sat’을 가지고 주변의 fat, cat, on, the 예측



III. New Log-linear Models

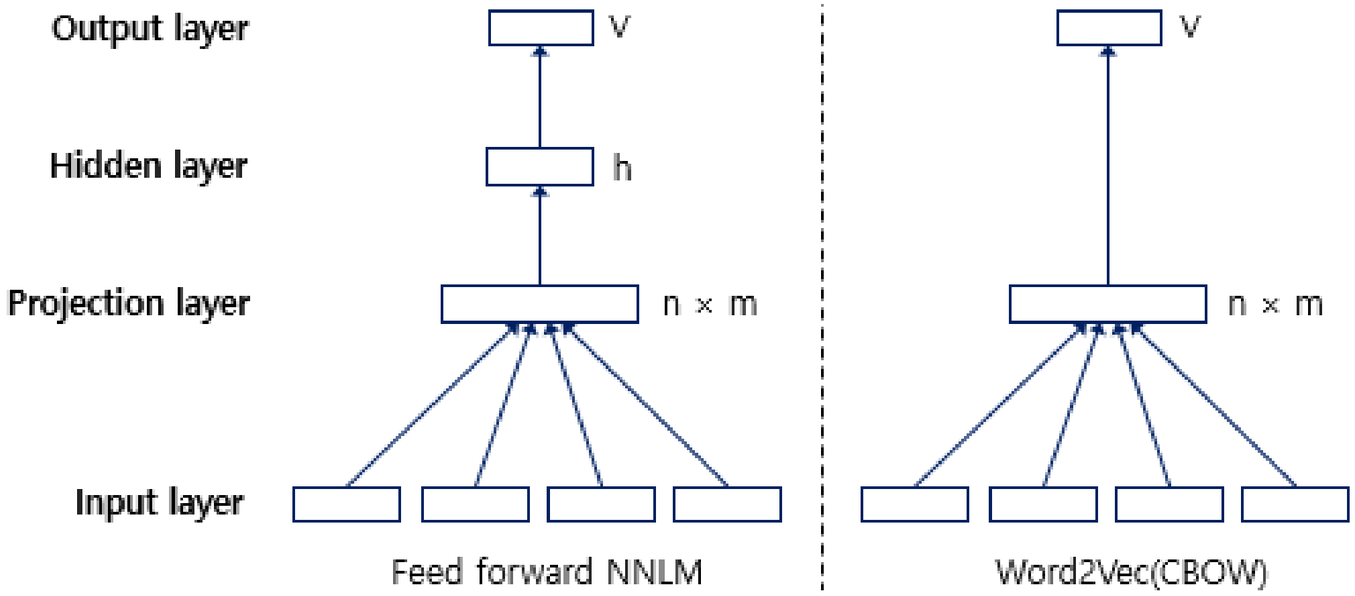
Word2Vec – skip-gram

‘sat’을 가지고 주변의 fat, cat, on, the 예측



III. New Log-linear Models

NNLM vs Word2Vec



n: 현재 학습 단어의 수

m: 임베딩 벡터 차원

h: 은닉층 크기

V: 단어 집합의 크기

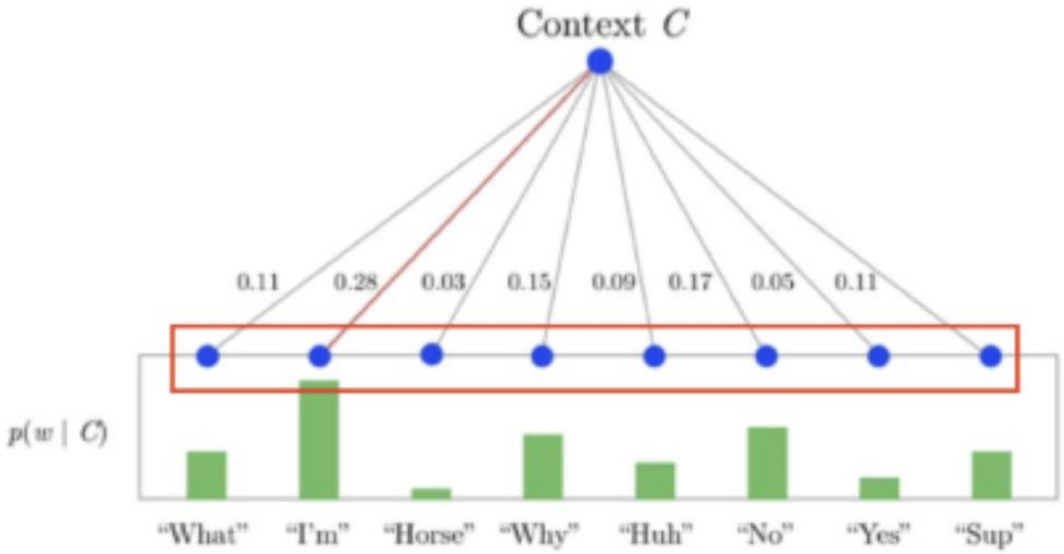
(n x m) + (n x m x h) + (h x V)

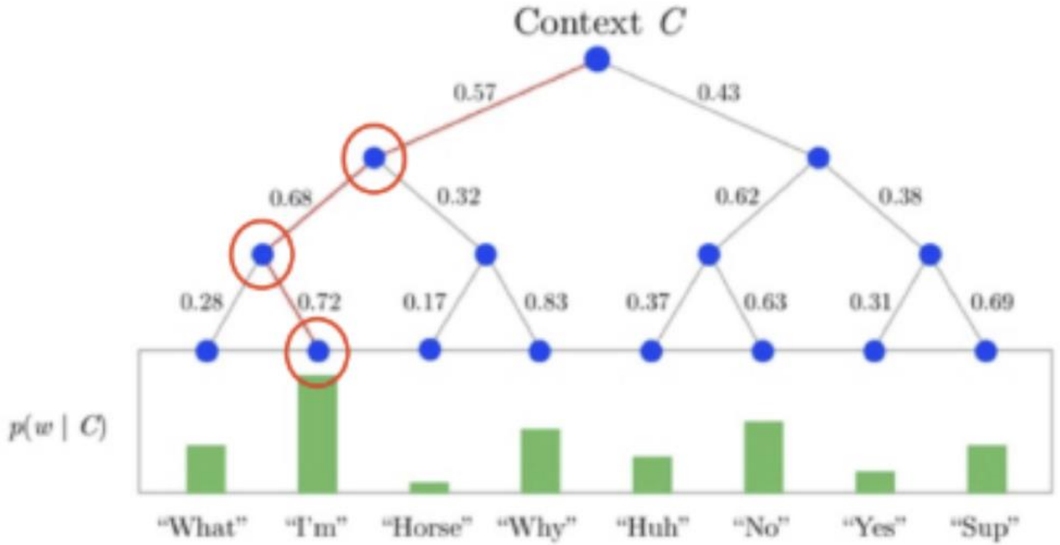
CBOW: (n x m) + (m x log(V))

Skip-gram: C x (m + m x log(V))

III. New Log-linear Models

NNLM vs Word2Vec -> Hierarchical Softmax





(n x m) + (n x m x h) + (h x V)

(n x m) + (m x log(V))

IV. Results

Big – biggest의 관계를 바탕으로

Small - ? 의 단어를 벡터 연산으로 구할 수 있음

Dimensionality / Training words	24M	49M	98M	196M	391M	783M
50	13.4	15.7	18.6	19.1	22.5	23.2
100	19.4	23.1	27.8	28.7	33.4	32.2
300	23.2	29.2	35.3	38.6	43.7	45.9
600	24.0	30.1	36.5	40.8	46.6	50.4

구글 뉴스 말뭉치 학습(단어의 개수와 임베딩 차원을 기준으로 학습 구분

IV. Results

Model Architecture	Semantic-Syntactic Word Relationship test set		MSR Word Relatedness Test Set [20]
	Semantic Accuracy [%]	Syntactic Accuracy [%]	
RNNLM	9	36	35
NNLM	23	53	47
CBOW	24	64	61
Skip-gram	55	59	56

기존 모델보다 word2vec이 높은 성능을 보였고,

CBOW는 Syntactic(구조적)

Skip-gram은 Semantic(의미적) 으로 우수

V. Examples of the Learned Relationships

Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

Paris – France + Italy = Rome

단어 목록의 평균 벡터를 계산하고, 가장 먼 단어 벡터를 찾음

-> 목록에서 벗어난 단어를 제외하는 태스크 수행 가능

VI. Conclusion

간단한 모델 아키텍처로 고품질의 단어 벡터 훈련

낮은 계산 복잡성으로 더 큰 데이터 세트에서 정확성 높임

기존의 NN 기반 임베딩 방법론이 NLP 태스크에 적용 되었는데,
이 논문의 방법론으로 더욱 효율적으로 적용 가능 할 것