Efficient Estimation of Word Representations in Vector Space

Language&AI 학회 Attention

2025.11.03 제1회 발제

발제자: 양수찬, 장지수

목차

- Introduction
- Previous Models (~2.3)
- CBOW Skipgram Structure
- 실험 결과
- Conclusion and Following Works

Introduction

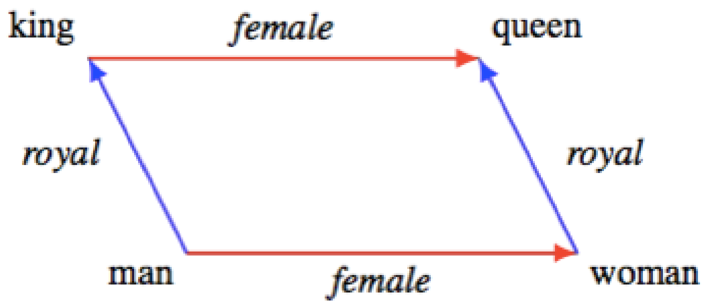
Previous NLP systems

- words = atomic units (symbolic, can not be calculated)
- N-gram model
- Limit 1: Leck of in-domain data for speech recognition
- Limit 2: Leck of existing corpora for machine translation

Introduction

Similarity of Word Vectors

- Words have <u>multiple degrees of similarity</u>(의미, 형태, 문법적 기능 등) and can be represented as <u>embedding vector</u>
- Word offset technique = algebraic operation with vectors
- Previous limit: <u>computationally expensive(complex)</u> → have to minimize



두 벡터의 관계가 거의 동일
vector(king) – vector(man) = vector(queen) – vector(woman)

이항
vector(king) – vector(man) + vector(woman) = vector(queen)

Introduction

Goals of Paper

- Maximize vector operation accuracy
- Preserve linear regularities among words
- How training time, accuracy depends on dimensionality of word vectors, amount of training data

Previous Models

Latent Semantic Analysis (LSA, 1990)

- 어떤 단어들이 어떤 문서에 함께 쓰였는지를 표로 만듦
- 단어 출현 빈도수에 따라 co-occurrence pattern을 찾음
- 행렬 계산을 통해 하나의 숨겨진(Latent) 차원으로 압축 → 같이 쓰이는 단어들을 하나의 차원으로 묶음
- 단어와 문서의 의미 관계 파악용

Latent Dirichlet Allocation (LDA, 2003)

- 단어의 조합 → 주제의 조합 → 문서 가정
- 문서의 조합을 확률적으로 역추정
- 통계 모델링 접근법

Previous Models

NNLM, RNNLM

Training complexity for NNLM, RNNLM (proportional)

O = E x T x Q

E: number of training epochs (3 ~ 50)

T: number of words in the training set (up to one billion)

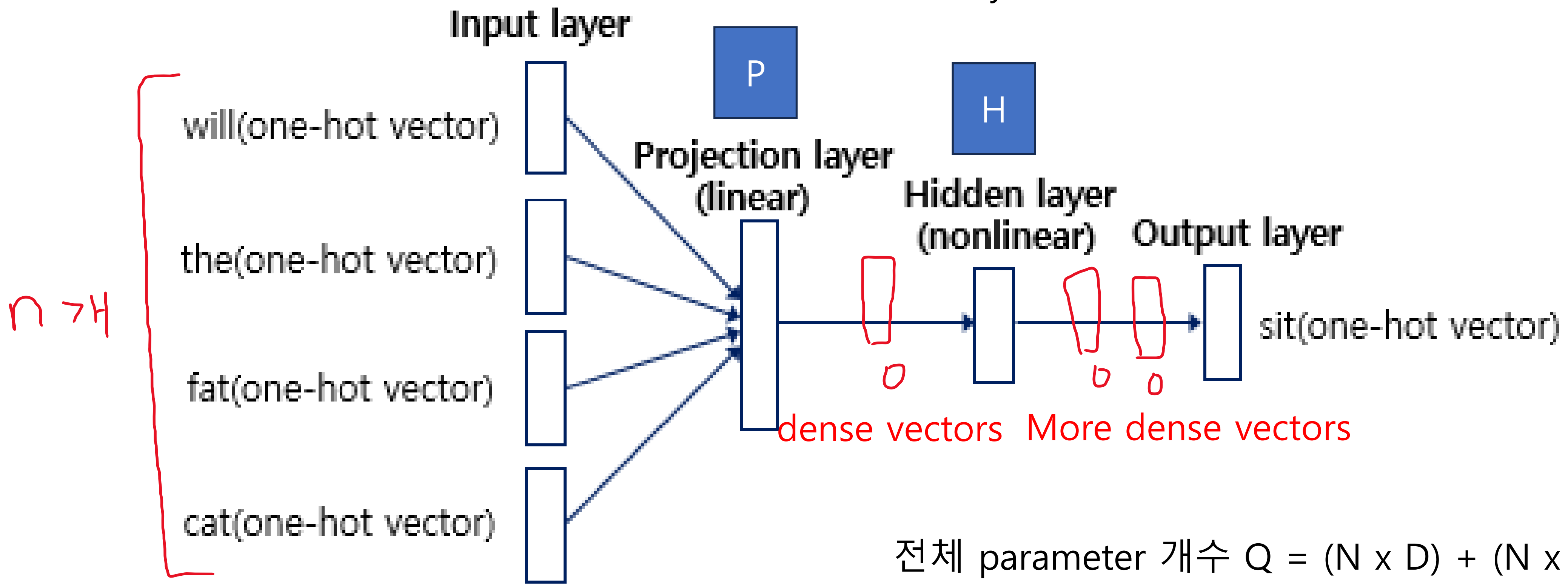
Q: we will define now!

Previous Models

Feedforward Neural Net Language Model (NNLM, 2003)

P의 차원: N x D

D is dimensionality of vector

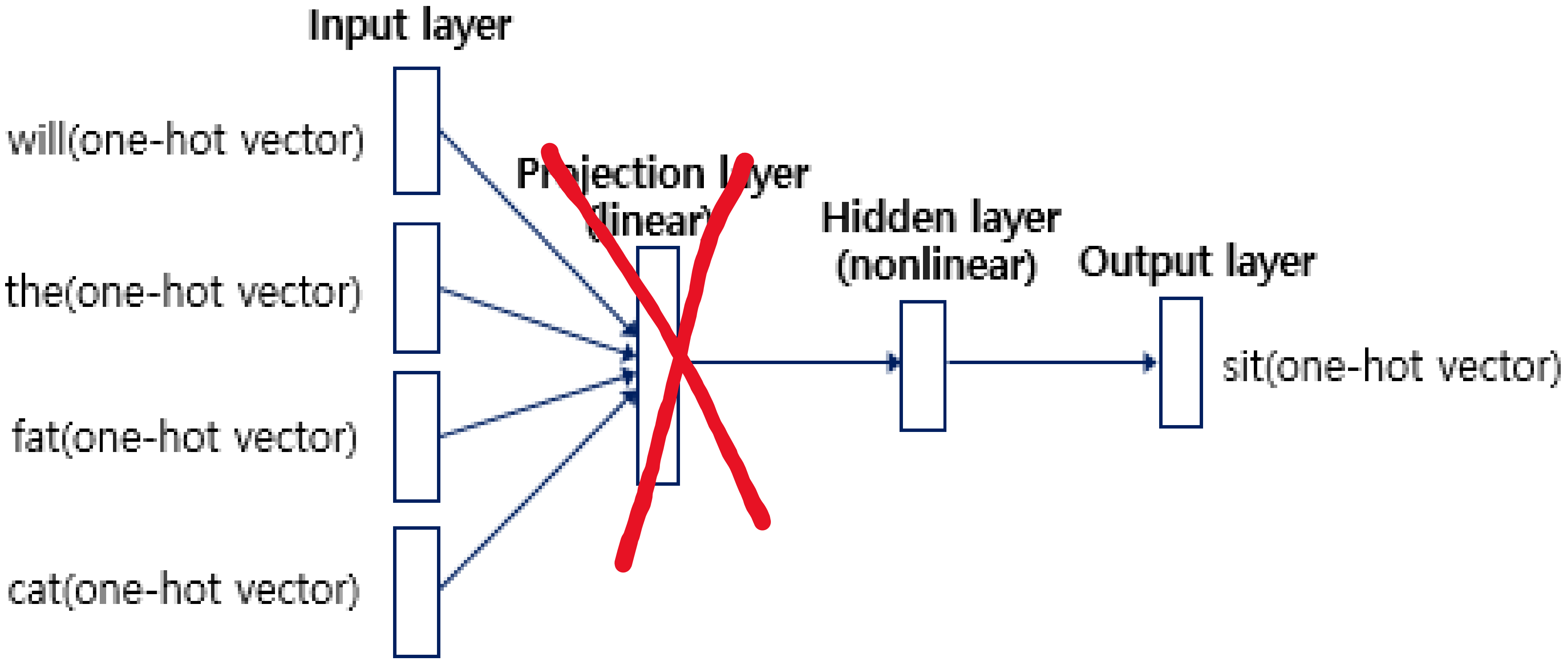


전체 parameter 개수 Q = (N x D) + (N x D x H) + (H x V)

Previous Models

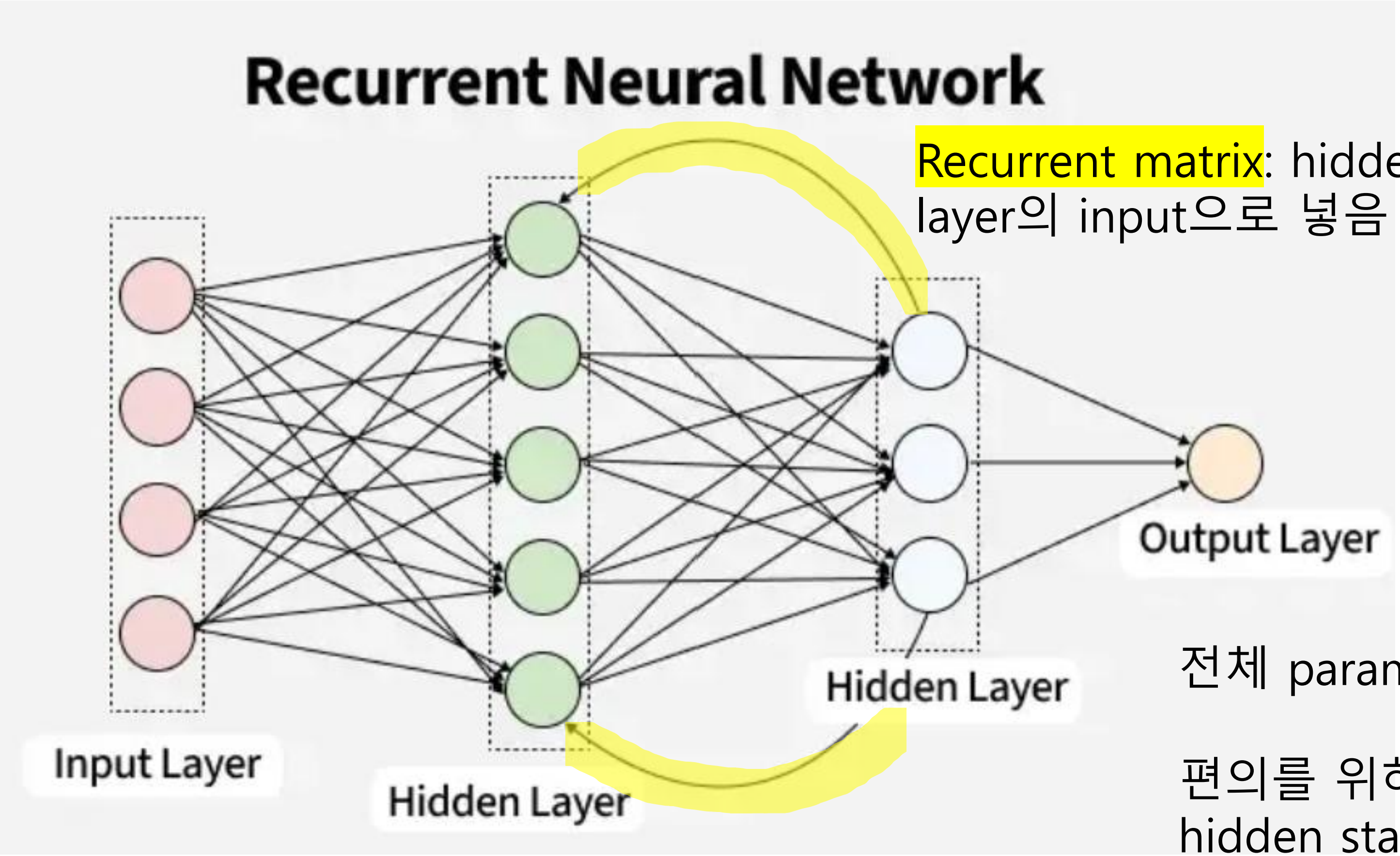
Recurrent Neural Net Language Model (RNNLM)

Can specify the context length



Previous Models

Recurrent Neural Net Language Model (RNNLM)



Recurrent matrix: hidden layer의 output을 다시 앞 layer의 input으로 넣음 → **short term memory**

전체 parameter 개수 Q = (H x H) + (H x V)

편의를 위하여 word embedding vector의 차원 D를 hidden state(H)의 차원과 같다고 가정

To avoid complexity in softmax function

- Hierarchical version of softmax
- Not normalized model

To avoid complexity in softmax function

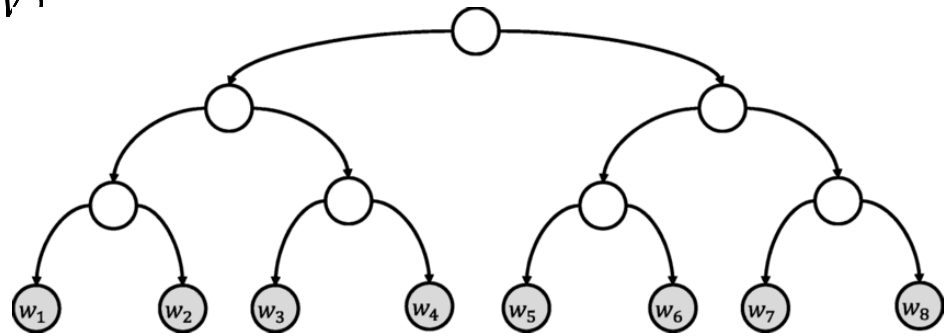
Hierarchical version of softmax

Balanced binary tree softmax: 단어의 등장 빈도와 상관없이 트리의 '높이(depth)' 최소화

전체 parameter 개수 Q = (N x D) + (N x D x H) + (H x $\log_2 Unigram\ perplexity\ V$)

Huffmann binary tree soft max: 자주 등장하는 단어를 root쪽에, 드문 단어를 leaf 쪽에 배치

전체 parameter 개수 Q = (N x D) + (N x D x H) + (H x $\log_2 V$)



To avoid complexity in softmax function

Not normalized model

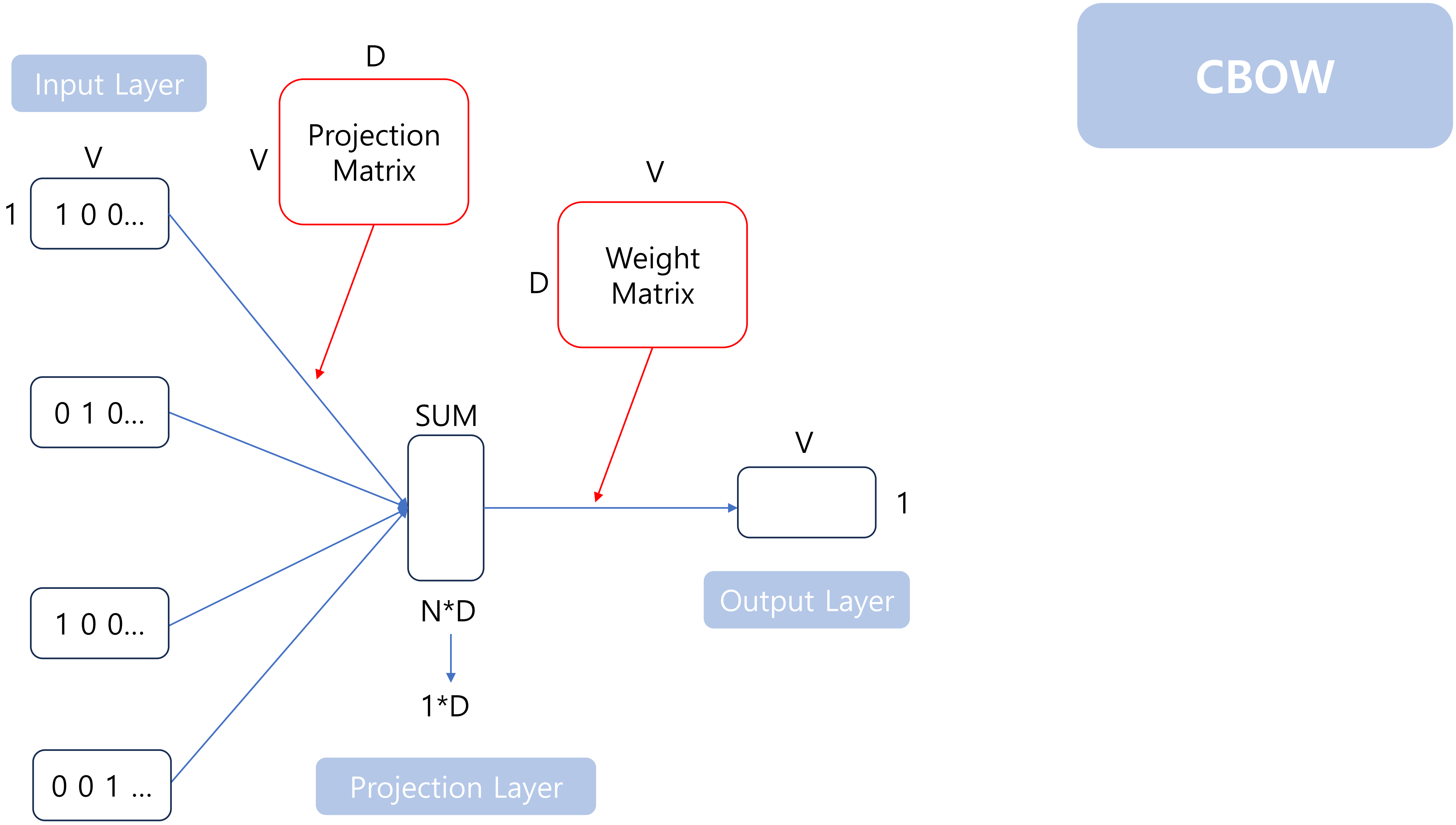
기존: 특정 단어 w가 정답일 확률 P계산 (multi class classificatoin)

not normalized: 특정 단어 w가 정답이냐 아니냐 계산 (binary)

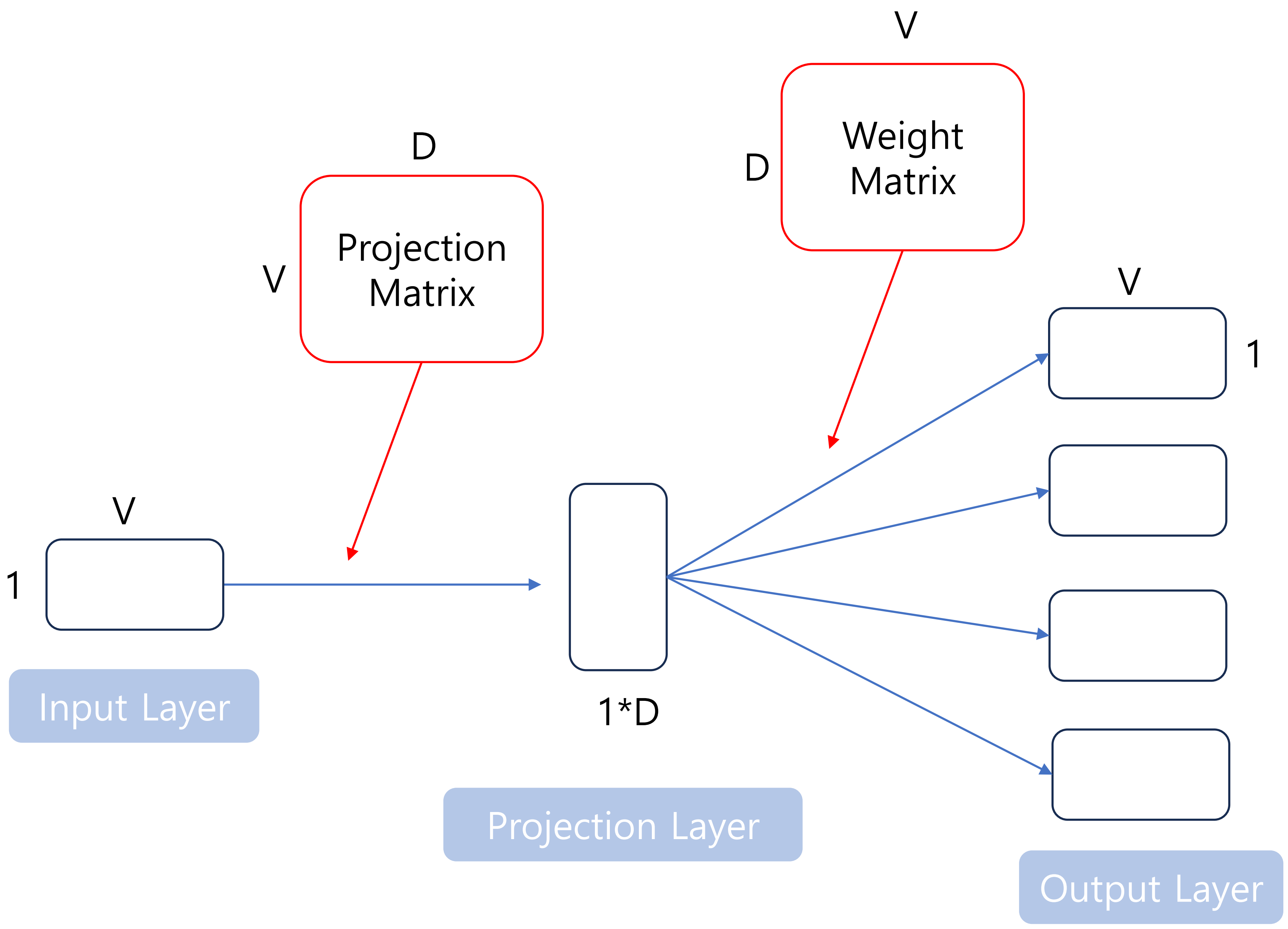
$$P(w) = \frac{\exp(\text{score}_w)}{\cancel{\sum_{j=1}^{V} \exp(\text{score}_j)}}$$

Model Training Framework

- **DistBelief**
- Runs multiple replicas of the same model in parallel
- All replicas synchronize their gradient updates (in server)
- Mini-batch asynchronous gradient descent with adaptive learing rate procedure(Adagrad) are used


CBOW
Input Layer
V
1
1 0 0...
0 1 0...
1 0 0...
0 0 1 ...
D
V
Projection Matrix
V
D
Weight Matrix
SUM
N*D
1*D
Projection Layer
V
1
Output Layer

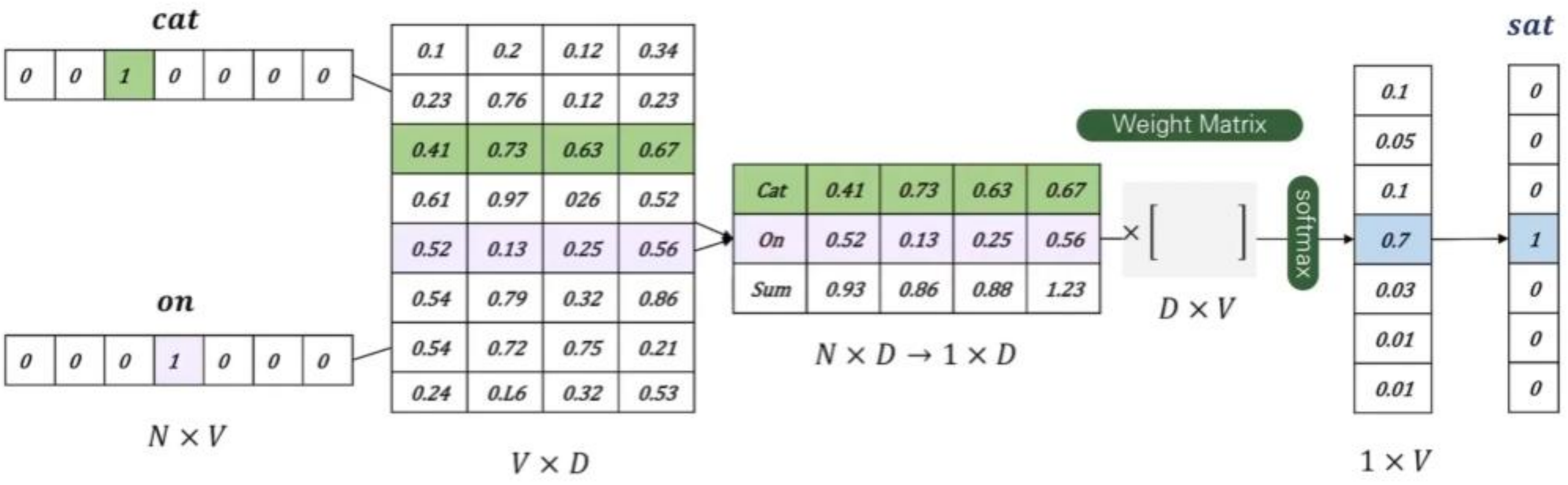
Skip-gram


V
D
Weight
Matrix
D
V
Projection
Matrix
V
1
V
1
1*D
Input Layer
Projection Layer
Output Layer

The cat sat on the mat.
Window size = 1

Example

- Example
  - ✓ “The fat cat sat on the mat”
    - – $window\ size = 1$



https://www.youtube.com/watch?v=sidPSG-EVDo&t=960

- **Computational Complexity**

  ✓ CBOW

  – $Q = N \times D + D \times V$

  1. $N \times D$: 현재 단어를 중심으로 N개의 단어 projection
  2. $D \times V$: projection layer에서 output layer 계산

  ✓ Skip-gram

  – $Q = C \times (D + D \times V)$

  1. $D + D \times V$: 현재 단어 projection + output 계산
  2. × C : C 개의 단어에 대해 진행해야 하므로 총 C배

https://www.youtube.com/watch?v=sidPSG-EVDo&t=960

Type of relationship	Word Pair 1		Word Pair 2	
Common capital city	Athens	Greece	Oslo	Norway
All capital cities	Astana	Kazakhstan	Harare	Zimbabwe
Currency	Angola	kwanza	Iran	rial
City-in-state	Chicago	Illinois	Stockton	California
Man-Woman	brother	sister	grandson	granddaughter
Adjective to adverb	apparent	apparently	rapid	rapidly
Opposite	possibly	impossibly	ethical	unethical
Comparative	great	greater	tough	tougher
Superlative	easy	easiest	lucky	luckiest
Present Participle	think	thinking	read	reading
Nationality adjective	Switzerland	Swiss	Cambodia	Cambodian
Past tense	walking	walked	swimming	swam
Plural nouns	mouse	mice	dollar	dollars
Plural verbs	work	works	speak	speaks

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

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

Model	Vector Dimensionality	Training words	Accuracy [%]			Training time [days]
			Semantic	Syntactic	Total	
3 epoch CBOW	300	783M	15.5	53.1	36.1	1
3 epoch Skip-gram	300	783M	50.0	55.9	53.3	3
1 epoch CBOW	300	783M	13.8	49.9	33.6	0.3
1 epoch CBOW	300	1.6B	16.1	52.6	36.1	0.6
1 epoch CBOW	600	783M	15.4	53.3	36.2	0.7
1 epoch Skip-gram	300	783M	45.6	52.2	49.2	1
1 epoch Skip-gram	300	1.6B	52.2	55.1	53.8	2
1 epoch Skip-gram	600	783M	56.7	54.5	55.5	2.5

Conclusion

- CBOW와 Skip-gram model을 사용하여 **기존 모델보다 더 simple한 모델로 더 높은 quality의 word vector 연산 가능**
- Much lower computational complexity → 고차원 계산 (더 많은 데이터 set)
- 1조개의 단어를 가진 corpora에서의 학습도 가능 (using DistBelief framework) → 이론상 무제한
- NLP tasks (sentiment analysis, paraphrase detection 등), Knowledge Base 확장, Machine Translation 분야 유망

Follow-up Work

- C++ code (CBOW, word2vec architecture 모두 사용)
- https://code.google.com/archive/p/word2vec/
- Also includes pre-trained word vectors (1000억개 단어로 학습된 140만개 이상의 named entit로 구성된 vectors)