UNIVERSITY OF SCIENCE & TECHNOLOGY

Korea Institute of Science and Technology Information

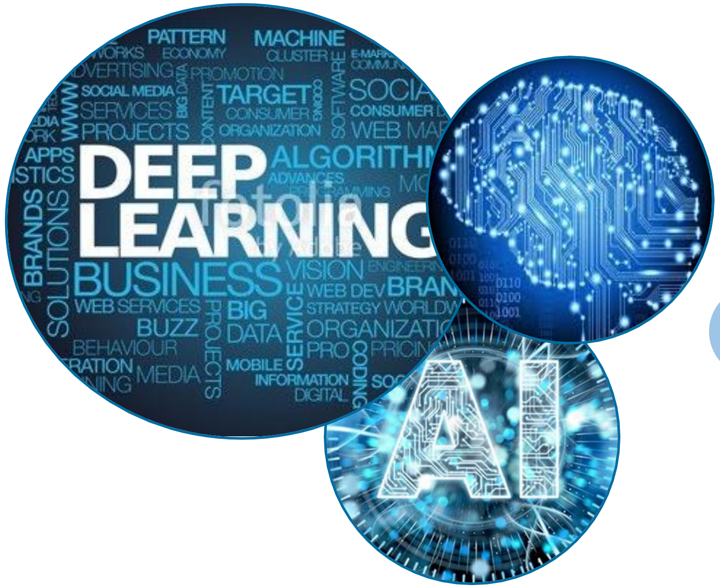
www.kisti.re.kr

2021-1 딥러닝기술 및 응용 - Paper Review

Distributed Representations of Words and Phrases and their Compositionality

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean

(Google Inc.)



KISTI-UST

Donghun Yang

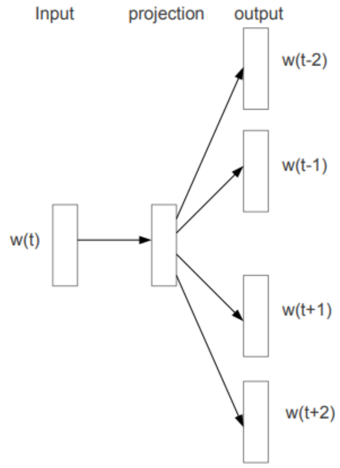
2021. 03. 19. FRI.

Contents

Abstract

The recently introduced continuous Skip-gram model is an efficient method for learning high-quality distributed vector representations that capture a large number of precise syntactic and semantic word relationships. In this paper we present several extensions that improve both the quality of the vectors and the training speed. By subsampling of the frequent words we obtain significant speedup and also learn more regular word representations. We also describe a simple alternative to the hierarchical softmax called negative sampling.

(around 2x - 10x speed up)



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

[Original Skip-gram Model]

$$\log\sigma({v'_{w_O}}^\top v_{w_I}) + \sum_{i=1}^{k}\mathbb{E}_{w_i\sim P_n(w)}\left[\log\sigma(-{v'_{w_i}}^\top v_{w_I})\right]$$

[Hierarchical Softmax]

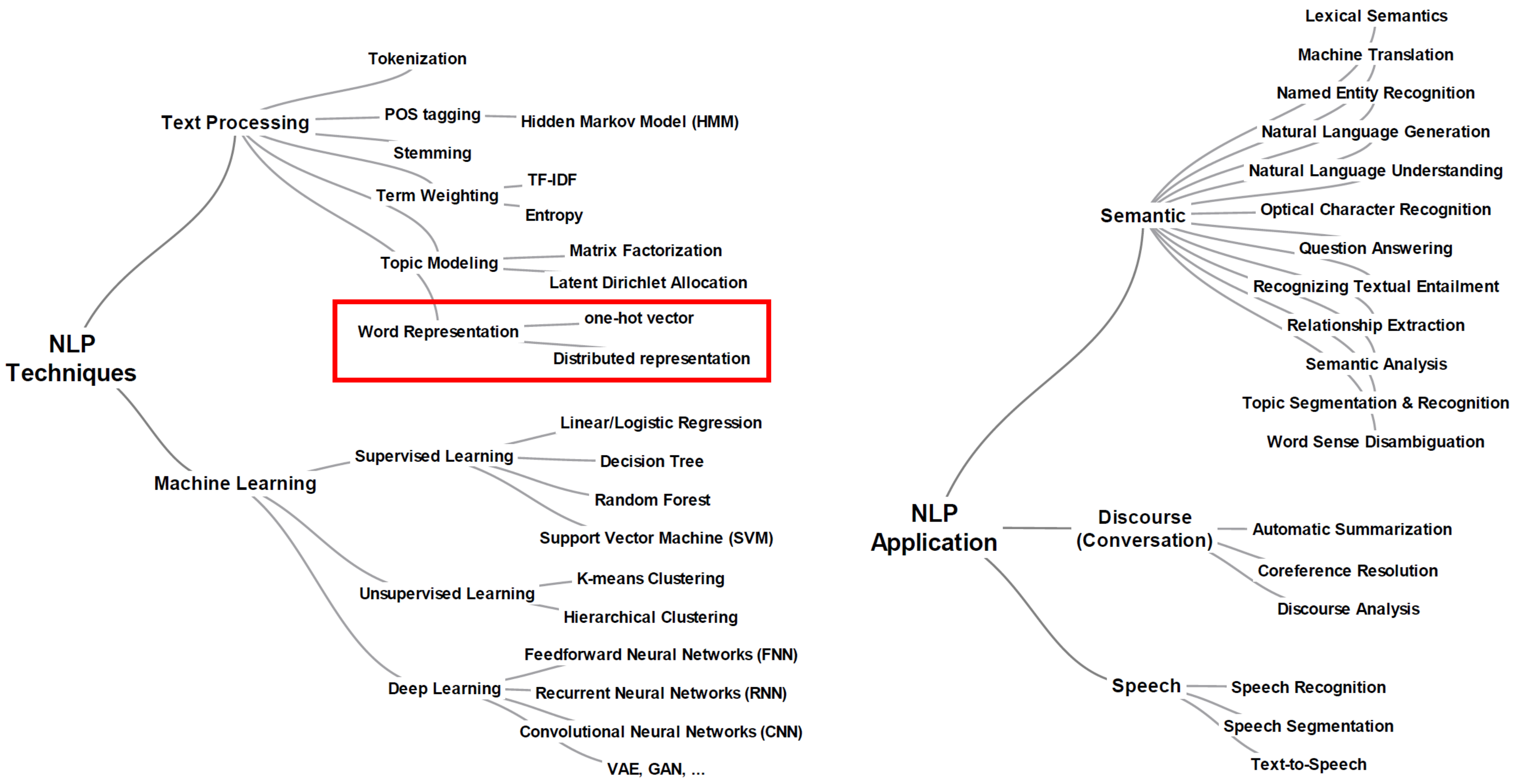
$$p(w|w_I) = \prod_{j=1}^{L(w)-1}\sigma\left([\![n(w,j+1) = \mathrm{ch}(n(w,j))]\!]\cdot {v'_{n(w,j)}}^\top v_{w_I}\right)$$

[Negative Sampling]

$$P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}}$$

[Subsampling of Frequent Words]

01 Introduction
Preliminary



[Category of NLP]

01 Introduction

Preliminary

✓ Word Representation : A method of representing words as vectors so that computers can understand and process natural language efficiently.

[Dog]

[Cat]

[Horse]

[Cow]

How To ?

01 Introduction

Preliminary

✓ Discrete Representation (Sparse Representation)

- Limitation : Curse of Dimensionality, Sparsity, Semantic

[Dog] [1 0 0 0]

[Cat] [0 1 0 0]

[Horse] [0 0 1 0]

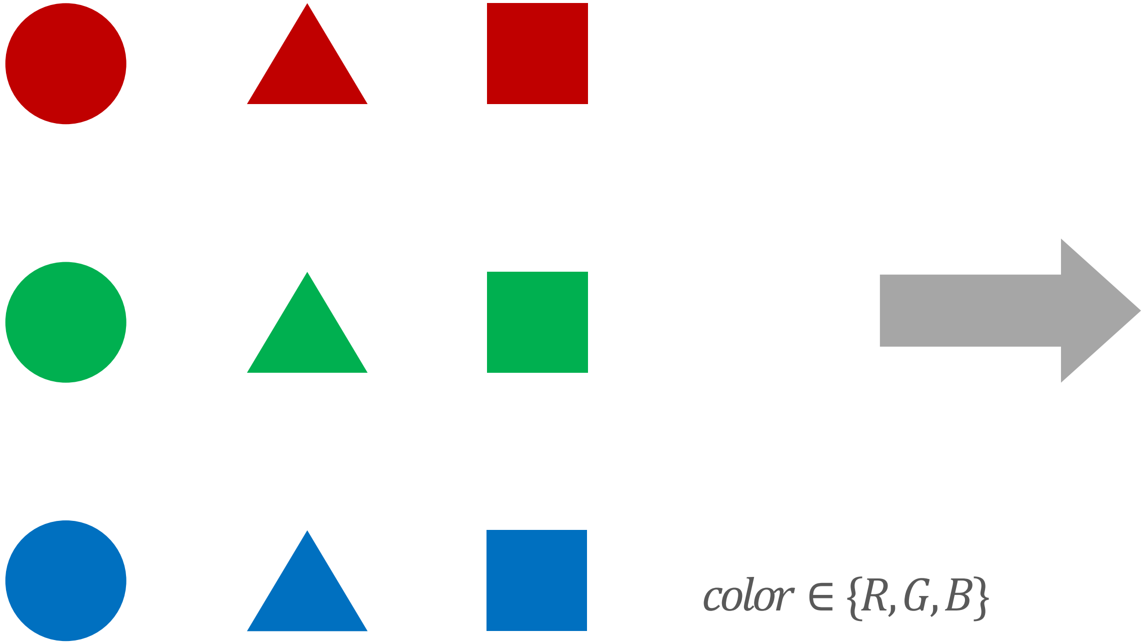
[Cow] [0 0 0 1]

01 Introduction

Preliminary

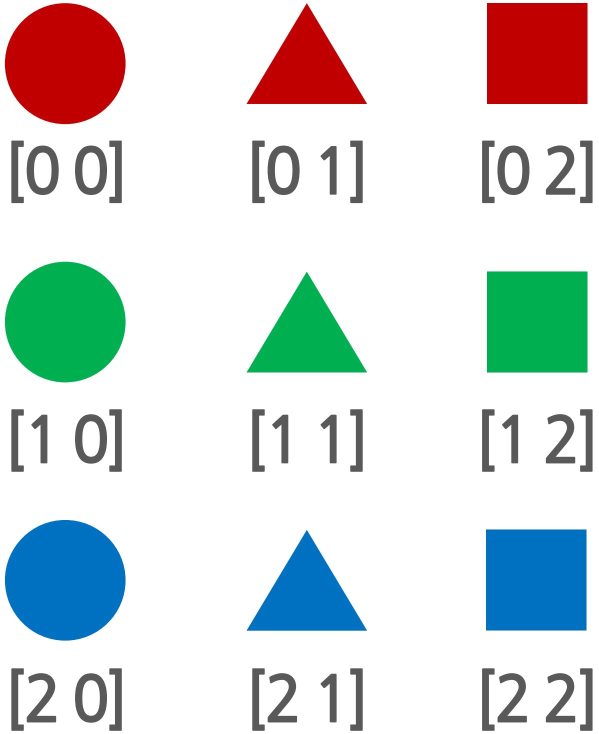
✓ **Distributed Representation (Dense Representation)**

- Similar words are located in similar position in vector space.



$color \in \{R, G, B\}$

$shape \in \{circle, triangle, rectangle\}$



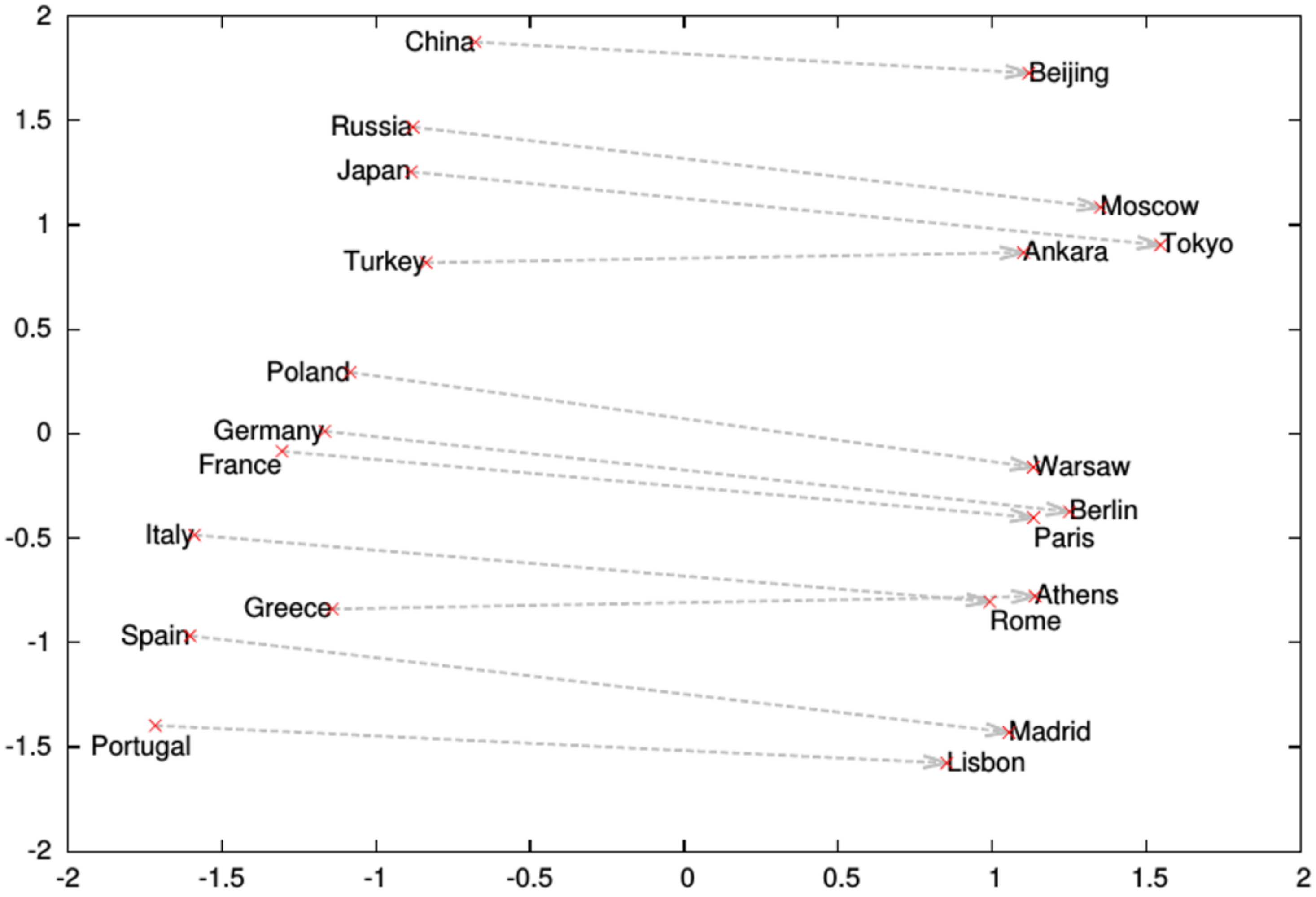
01 Introduction

Preliminary

✓ Sparse Representation V.S. Distributed Representation

		Sparse Representation	Distributed Representation
	[Dog]	[1 0 0 0 0]	[9 0 0 0]
	[Cat]	[0 1 0 0 0]	[9 1 0 0]
	[Horse]	[0 0 1 0 0]	[1 0 1 9]
	[Cow]	[0 0 0 1 0]	[1 0 9 1]
	[rion]	[0 0 0 0 1]	[1 9 0 1]

01 Introduction
Preliminary



01 Introduction

Previous Study - NPLM (Bengio 2003)

"발 없는 말이 천리 간다"

4–gram

발, 없는, 말이, ??(t=4)

없는, 말이, 천리, ??(t=5)

✓ Goal : Predicting the target word using the previous n-1 words

Step 1 : Initialize C (Shared-Lookup table)

$$C = \begin{array}{c} \begin{bmatrix} 11 & 18 & 25 \\ 10 & 12 & 19 \\ 4 & 6 & 13 \\ 23 & 5 & 7 \\ 17 & 24 & 1 \end{bmatrix} \end{array} \begin{array}{l} \text{발} \\ \text{없는} \\ \text{말이} \\ \text{천리} \\ \text{간다} \end{array}$$

Step 2 : Dot product One-hot Vector of each word and C

$$\overset{\text{말이}}{[\,0\ 0\ 0\ 1\ 0\,]} \times \begin{bmatrix} 11 & 18 & 25 \\ 10 & 12 & 19 \\ 4 & 6 & 13 \\ 23 & 5 & 7 \\ 17 & 24 & 1 \end{bmatrix} = \overset{x_{\text{말이}}}{[\,23\ \ 5\ \ 7\,]}$$

Step 3 : Concatenate all x

$$x = [x_{t-1}, x_{t-2}, \ldots, x_{t-n+1}]$$

$$x = [\underbrace{10, 12, 19}_{\text{발}}, \underbrace{4, 6, 13}_{\text{없는}}, \underbrace{23, 5, 7}_{\text{말이}}]$$

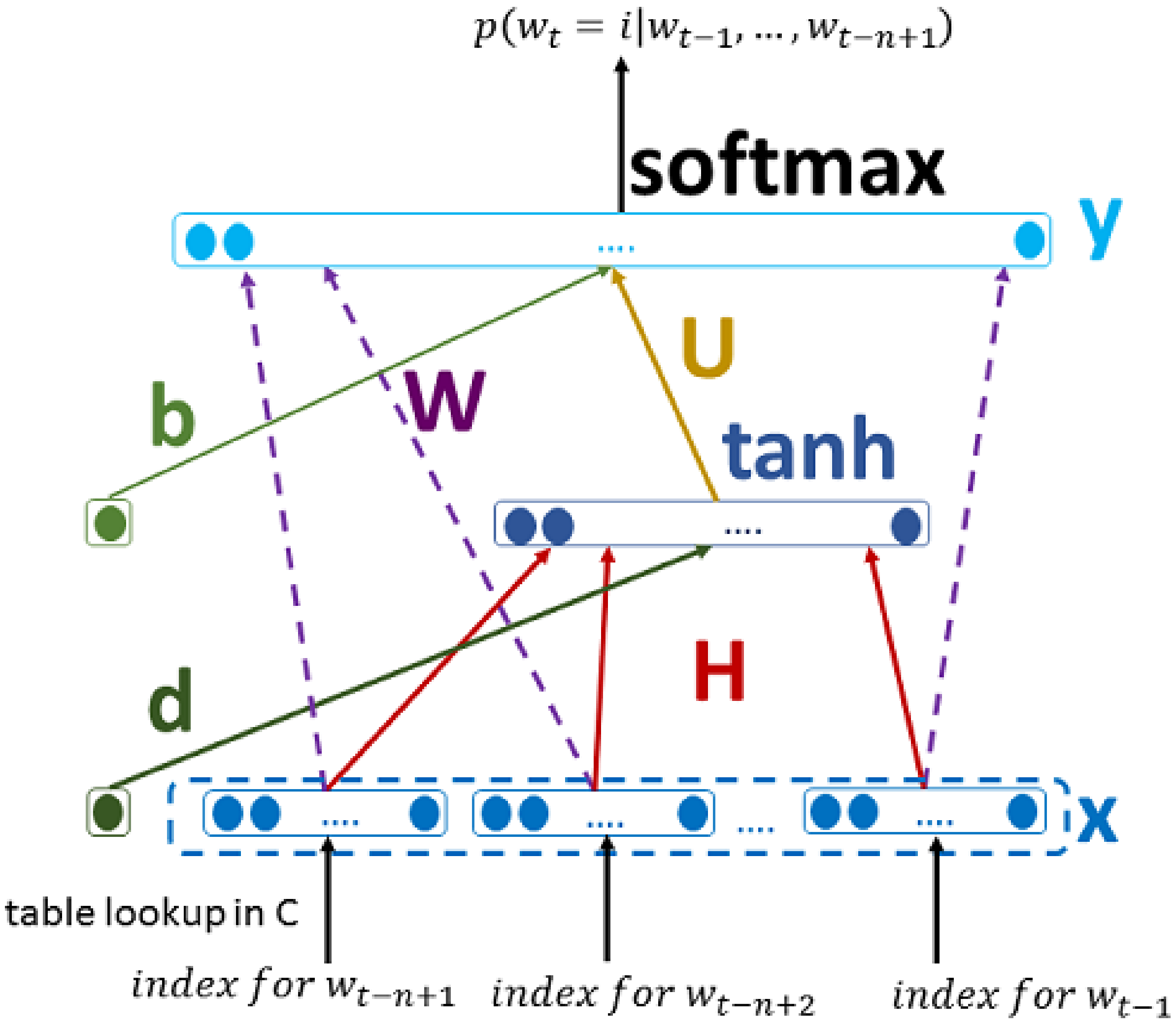
Step 4 : $y_{w_t} = b + Wx + Utanh(d + Hx)$

Step 5 : Apply softmax function to y_{w_t}

$$P(w_t | w_{t-1}, \ldots, w_{t-n+1}) = \frac{exp(y_{w_t})}{\sum_i exp(y_i)}$$

01 Introduction

Previous Study - NPLM (Bengio 2003)



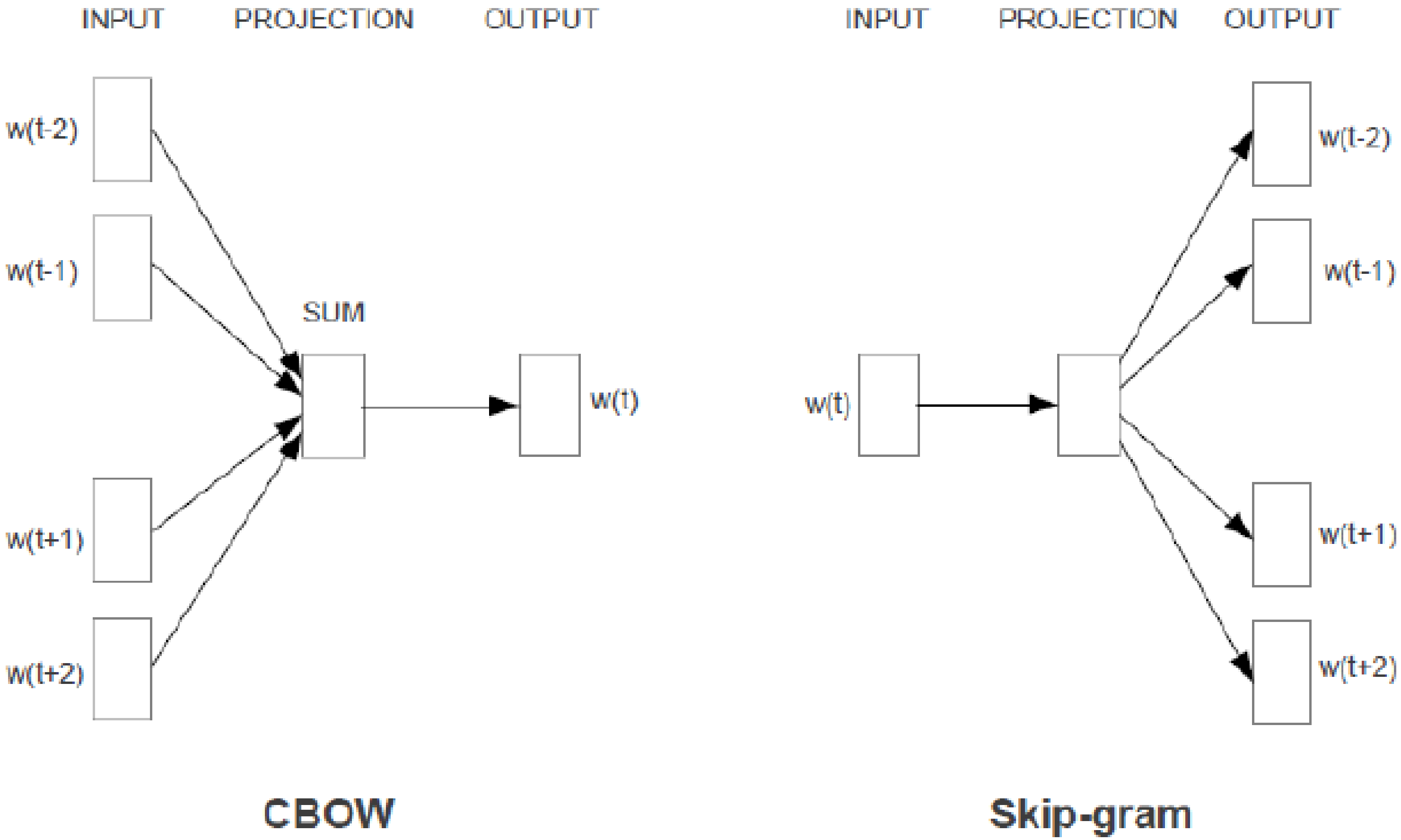
✓ Limitation : Too many parameters and computations

$$H \in R^{h\times(n-1)m}, \quad x_t \in R^{(n-1)\times m}, \quad d \in R^{h\times 1}$$

$$U \in R^{|V|\times h}, \quad b \in R^{|V|}, \quad y \in R^{|V|} \quad , C \in R^{m\times|V|}$$

01 Introduction

Previous Study - word2vec (Mikolov 2013)

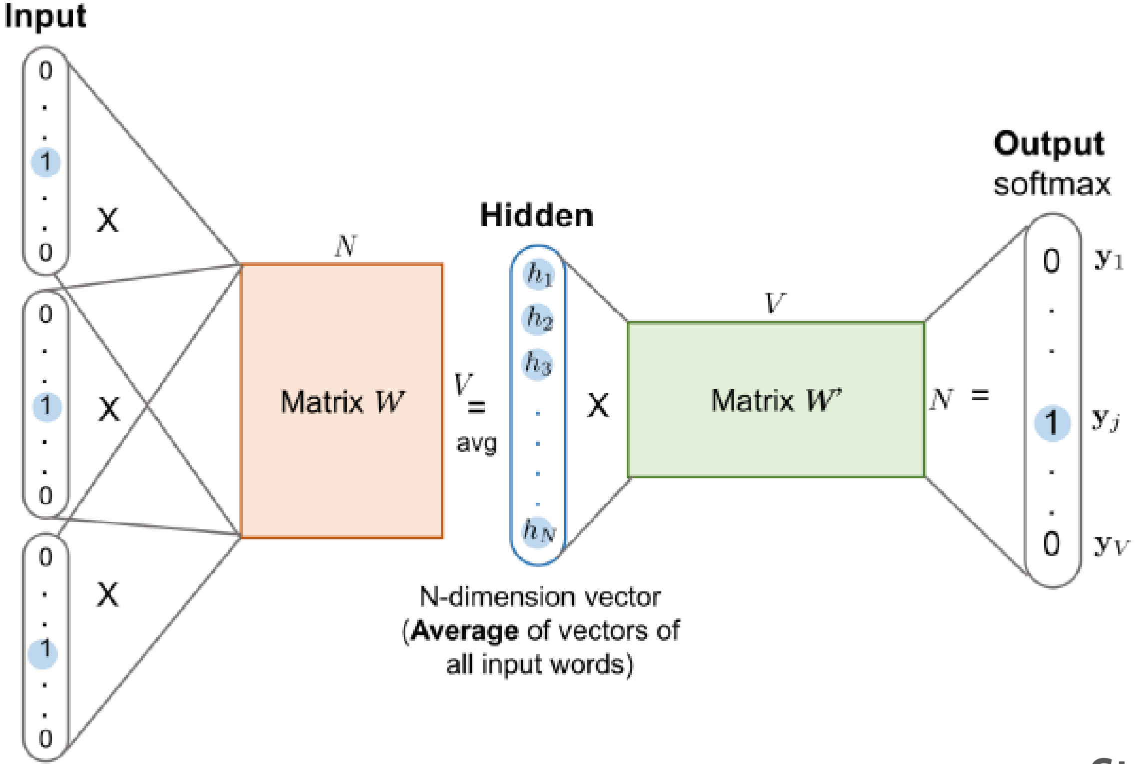


01 Introduction

Previous Study - word2vec (Mikolov 2013) - CBOW

✓ **CBOW**

- Predicting target word using context word
- Goal : Learning word representation



e.g. "The fat cat sat on the mat", n = 2

X = [fat cat on the], y = sat

Step 1 :

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

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0.5 & 2.1 & 1.9 & 1.5 & 0.8 \\ 0.8 & 1.2 & 2.8 & 1.8 & 2.1 \\ 2.1 & 1.8 & 1.5 & 1.7 & 2.7 \\ & & & & \\ & & & & \\ & & & & \\ & & & & \end{bmatrix} = \begin{bmatrix} 2.1 & 1.8 & 1.5 & 1.7 & 2.7 \end{bmatrix}$$

lookup table

Step 2 :

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

Step 3 : $z = v * w'$

Step 4 : Y = Softmax(z)

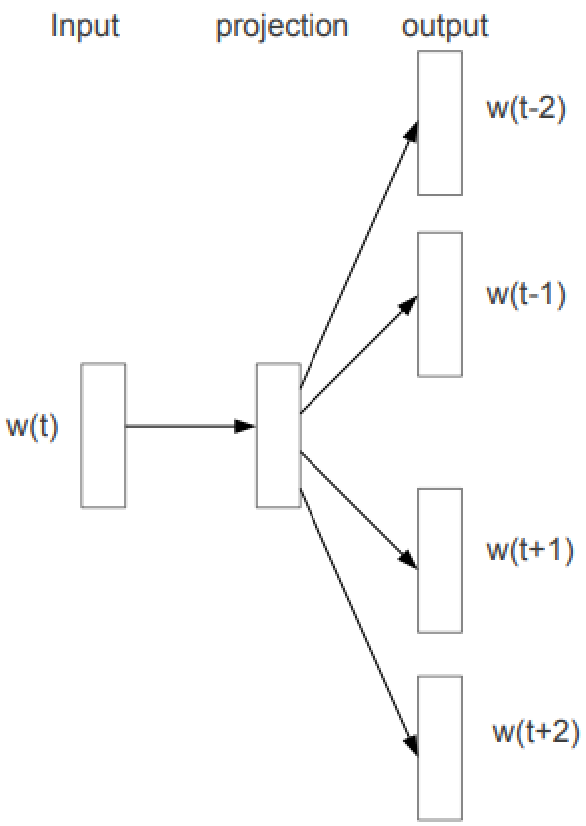
Step 5 : Minimize loss

$$\begin{aligned} minimize J &= -\log P(w_c|w_{c-m},\ldots,w_{c+m}) \\ &= -\log P(u_c|v) \\ &= -\log \frac{exp(u_c^\intercal \hat{v})}{\sum_{j=1}^{|V|} exp(u_j^\intercal \hat{v})} \\ &= -u_c^{intercal}\hat{v} + \log\sum_{j=1}^{|V|} exp(u_j^\intercal \hat{v}) \end{aligned}$$

Step 6 : Use W, W′ , W|W′ or (W+W′)/2 as word representation

02 The Skip-gram Model

Original Skip-gram Model



[Original Skip-gram Model]

- ✓ **Goal : To find word representations that are useful for predicting the surrounding words in a sentence or a document**
- ✓ **Objective Function :**

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

c : The size of the training context
w_t : The center word

$$p(w_O|w_I) = \frac{\exp\left({v'_{w_O}}^{\top} v_{w_I}\right)}{\sum_{w=1}^{W}\exp\left({v'_w}^{\top} v_{w_I}\right)}$$

w_O : The output word
w_I : The input word
v' : The vector representations of output word
v : The vector representations of input word
W : The number of words in vocabulary

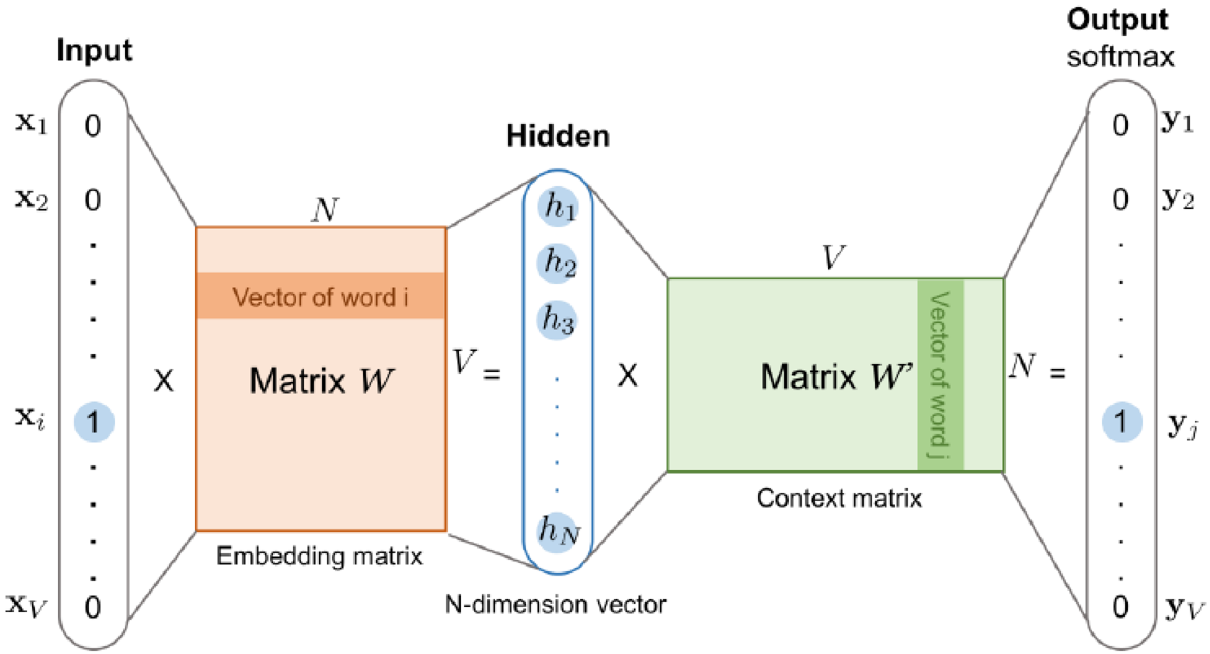
Complexity : $O(V)$

02 The Skip-gram Model

Original Skip-gram Model

e.g. “The fat cat sat on the mat”, n=2

(x, y) = (sat, fat), (sat, cat), (sat, on), (sat, the)



Step 1 : $x_{sat} \times W_{V\times M} = v_{sat}$

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
0.5	2.1	1.9	1.5	0.8
2.1	1.8	1.5	1.7	2.7

=

2.1	1.8	1.5	1.7	2.7

lookup table

~~**Step 2 :**~~ $\cancel{v = \frac{V_{fat}+V_{cat}+V_{on}+V_{the}}{2\times n(window\ size)}}$

Step 3 : z = v * w′

Step 4 : Y = Softmax(z)

Step 5 : Minimize loss

$$\text{minimize } J = -\log P(w_{c-m},\dots,w_{c-1},w_{c+1},\dots,w_{c+m}|w_c)$$
$$= -\log \prod_{j=0,j\neq m}^{2m} P(w_{c-m+j}|w_c)$$
$$= -\log \prod_{j=0,j\neq m}^{2m} \frac{\exp(u_{c-m+j}^{T} v_c)}{\sum_{k=1}^{|V|}\exp(u_k^{T} v_c)}$$
$$= -\sum_{j=0,j\neq m}^{2m} u_{c-m+j}^{T} v_c + 2m\log\sum_{k=1}^{|V|}\exp(u_k^{T} v_c)$$

Step 6 : Use W, W′ , W|W′ or (W+W′)/2 as word representation

02 The Skip-gram Model

NPLM V.S. CBOW V.S. Original Skip-gram Model

✓ The number of parameters is reduced.

$$H \in R^{h\times(n-1)m}, \quad x_t \in R^{(n-1)\times m}, \quad d \in R^{h\times 1}$$
$$U \in R^{|V|\times h}, \quad b \in R^{|V|}, \quad y \in R^{|V|} \quad , C \in R^{m\times|V|}$$

[NPLM]

→

$$W \in R^{|V|\times N}, W' \in R^{N\times|V|}$$

[CBOW and Skip-gram]

2*w x N + N x V

[CBOW]

(1 x N + N x V)*2*w

[Skip-gram]

✓ But it's still too expensive because of the softmax function. (Complexity O(V))

02 The Skip-gram Model

Hierarchical Softmax

✓ Goal : To computationally efficient approximate the full softmax

$$p(w|w_I) = \prod_{j=1}^{L(w)-1} \sigma\left([\![n(w,j+1) = \mathrm{ch}(n(w,j))]\!] \cdot {v'_{n(w,j)}}^{\top} v_{w_I} \right)$$

$n(w,j)$: The j - th node on the path from the root to w

$L(w)$: The length of above path

$n(w,1)$: The root

$n(w,L(w))$: w

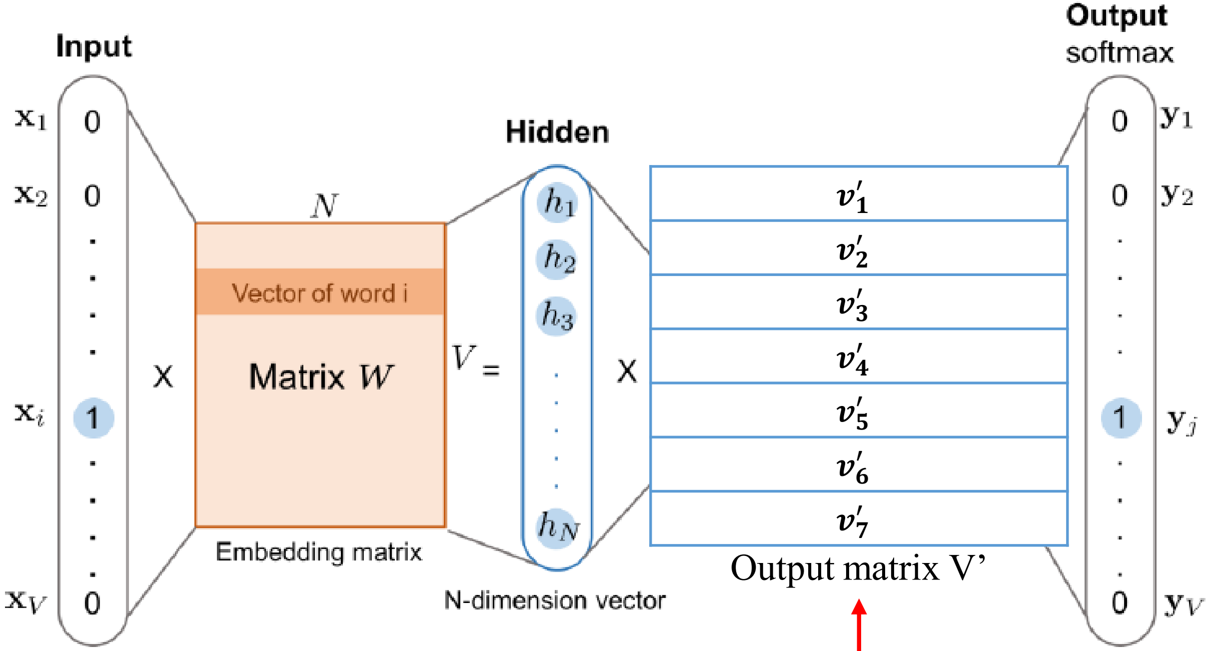
$ch(n)$: The left child of n

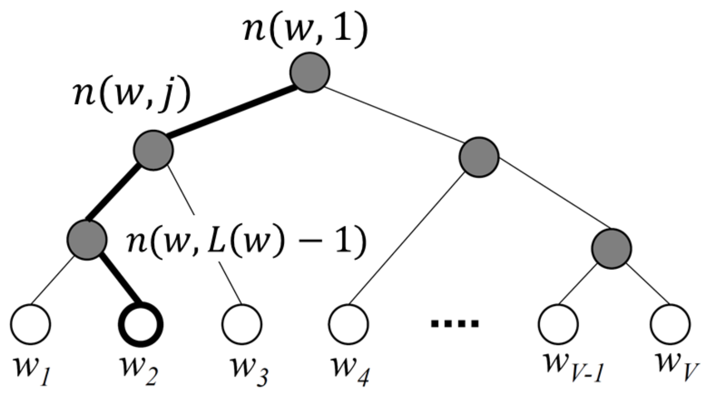
$[\![x]\!]$: 1 if x is true (left), -1 otherwise

$\sigma(x) = \frac{1}{1+exp(-x)}$: Sigmoid Function

$\sum_{w=1}^{W} p(w|w_i) = 1$

Complexity : $O(log_2 V)$





02 The Skip-gram Model

Hierarchical Softmax

Step 1 : Initialize output tree and matrix (V′)

Step 2 :

$$\boldsymbol{x}_{cat} \times \boldsymbol{W}_{V\times M} = \boldsymbol{V}_{cat}$$

$\boldsymbol{x}_{cat}$:

0	0	1	0	0	0	0

×

$\boldsymbol{W}_{V\times M}$:

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

=

$\boldsymbol{V}_{cat}$:

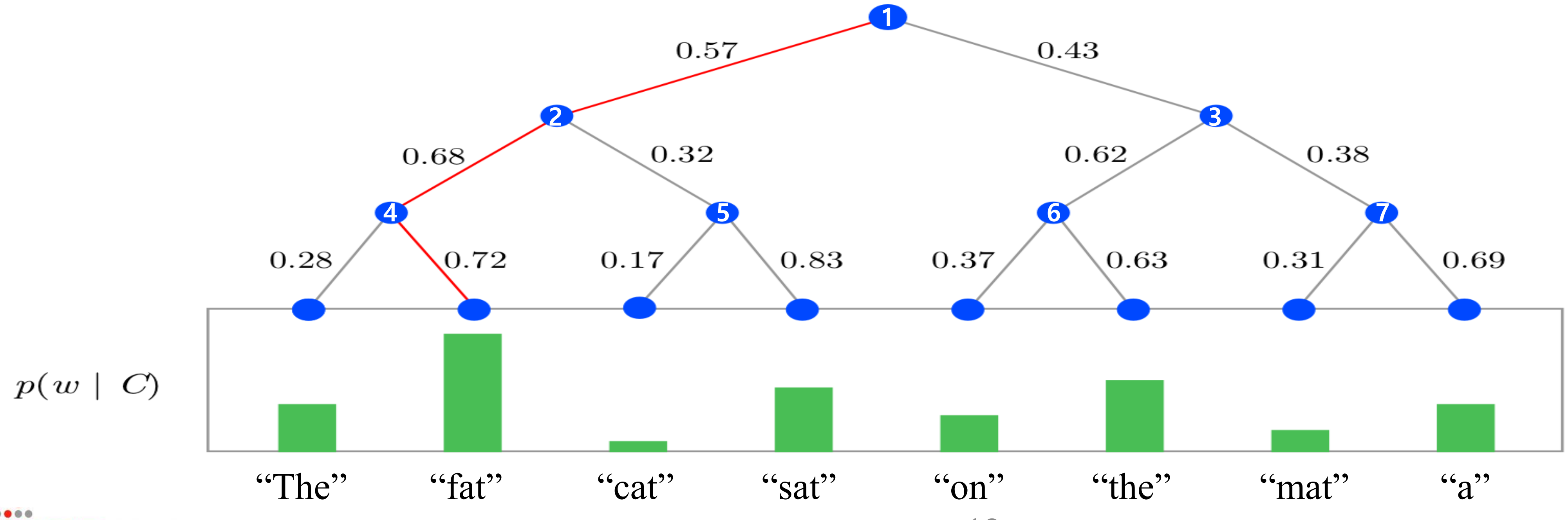
2.1	1.8	1.5	1.7	2.7

lookup table

Step 3 : $$p(w|w_I) = \prod_{j=1}^{L(w)-1} \sigma\left(\llbracket n(w,j+1) = \mathrm{ch}(n(w,j)) \rrbracket \cdot {v'_{n(w,j)}}^{\top} v_{w_I}\right)$$

e.g.

$$\begin{aligned} p(w_{fat}|w_{cat}) &= \sigma(\llbracket n(w_2,2) = ch(n(w_2,1)) \rrbracket \cdot {v'_{n(w_2,1)}}^{\top} v_{w_I}) \\ &\quad \times \sigma(\llbracket n(w_2,3) = ch(n(w_2,2)) \rrbracket \cdot {v'_{n(w_2,2)}}^{\top} v_{w_I}) \\ &\quad \times \sigma(\llbracket w_2 = ch(n(w_2,3)) \rrbracket \cdot {v'_{n(w_2,3)}}^{\top} v_{w_I}) \\ &= \sigma(1 \cdot {v'_{n(w_2,1)}}^{\top} v_{w_I}) \times \sigma(1 \cdot {v'_{n(w_2,2)}}^{\top} v_{w_I}) \times \sigma((-1) \cdot {v'_{n(w_2,3)}}^{\top} v_{w_I}) \\ &= 0.57 \times 0.68 \times 0.72 \end{aligned}$$



v'_1
v'_2
v'_3
v'_4
v'_5
v'_6
v'_7

Output matrix V'

02 The Skip-gram Model

Negative Sampling

- ✓ Goal : To computationally efficient approximate the full softmax
- ✓ Change problem setting multi classification to binary classification (+, -)

For Positive Sample — For Negative Sample

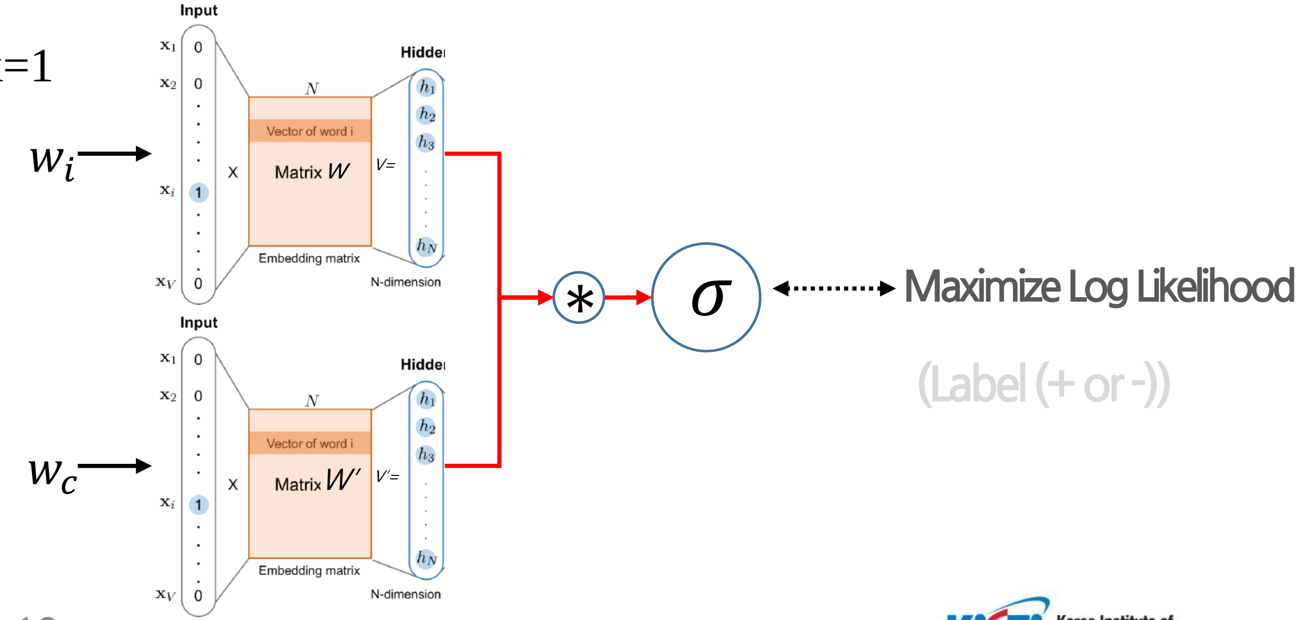
$$\log \sigma({v'_{w_O}}^{\top} v_{w_I}) + \sum_{i=1}^{k} \mathbb{E}_{w_i \sim P_n(w)} \left[\log \sigma(-{v'_{w_i}}^{\top} v_{w_I}) \right]$$

k : The number of negative sample (5-20 for small datasets, 2-5 for large datasets)

e.g. "The fat cat sat on the mat that can be tee of toy ", n=2, k=1

			[Positive Set]	[Negative Set]
(x, y) =	(sat, fat),	(x1, x2) =	(sat, fat),	(sat, tee),
	(sat, cat),		(sat, cat),	(sat, toy),
	(sat, on),		(sat, on),	(sat, mat),
	(sat, the)		(sat, the)	(sat, can)

[Original Skip-gram] [Skip-gram with Negative Sampling]



02 The Skip-gram Model
Subsampling of Frequent Words

✓ **Goal : To counter the imbalance between the rare and frequent words**

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

e.g. "A, A, B"

$$f(A) = p(A) = \frac{2}{3} \qquad f(B) = p(B) = \frac{1}{3}$$

$$P(A) = \frac{\left(\frac{2}{3}\right)^{\frac{3}{4}}}{\left(\frac{2}{3}\right)^{\frac{3}{4}} + \left(\frac{1}{3}\right)^{\frac{3}{4}}} = 0.6271$$

$$P(B) = \frac{\left(\frac{1}{3}\right)^{\frac{3}{4}}}{\left(\frac{2}{3}\right)^{\frac{3}{4}} + \left(\frac{1}{3}\right)^{\frac{3}{4}}} = 0.3729$$

[The probability of Negative sampling]

$$P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}}$$

t : Hyper Parameter (10^{-5})

- Calculate the probability of being excluded in training

e.g. "A, A, B"

$$f(A) = p(A) = \frac{2}{3} \qquad f(B) = p(B) = \frac{1}{3}$$

$$P(A) = 1 - \sqrt{\frac{10^{-5}}{\left(\frac{2}{3}\right)}} = 0.9961$$

$$P(B) = 1 - \sqrt{\frac{10^{-5}}{\left(\frac{1}{3}\right)}} = 0.9945$$

[The probability of Subsampling]

02 The Skip-gram Model

Code Review

✓ We will implement Word2Vec. (Skip-gram)

1. Build corpus from nltk brown.
2. Implement subsampling & Make vocabulary.
3. Building bag of words.
4. Implement network.
5. Training network.
6. Check original skip-gram result.
7. Implement skip-gram with negative sampling.

03 Empirical Results
Empirical Results

✓ **To evaluate the Hierarchical Softmax (HS), Noise Contrastive Estimation, Negative Sampling, and subsampling of the training words, the analogical reasoning task is used.**

- Datasets : Large dataset consisting of various news articles(an internal Google dataset with one billion words)
  : All words that occurred less than 5 times in the training data is discarded from the vocabulary.
  (Final vocabulary of size 692K)

Ex) Vec("Berlin") - Vec("Germany") + Vec("France") = Vec("Paris")

Method	Time [min]	Syntactic [%]	Semantic [%]	Total accuracy [%]
NEG-5	38	63	54	59
NEG-15	97	63	58	**61**
HS-Huffman	41	53	40	47
NCE-5	38	60	45	53
The following results use 10^{-5} subsampling				
NEG-5	14	61	58	60
NEG-15	36	61	61	**61**
HS-Huffman	21	52	59	55

Table 1: Accuracy of various Skip-gram 300-dimensional models on the analogical reasoning task as defined in [8]. NEG-k stands for Negative Sampling with k negative samples for each positive sample; NCE stands for Noise Contrastive Estimation and HS-Huffman stands for the Hierarchical Softmax with the frequency-based Huffman codes.

03 Empirical Results

Empirical Results

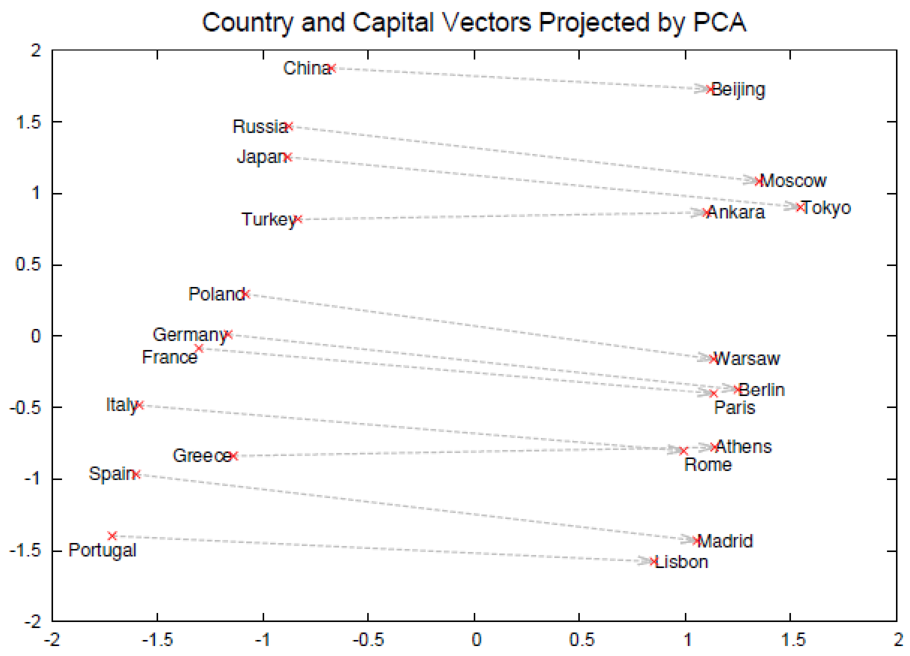


Figure 2: Two-dimensional PCA projection of the 1000-dimensional Skip-gram vectors of countries and their capital cities. The figure illustrates ability of the model to automatically organize concepts and learn implicitly the relationships between them, as during the training we did not provide any supervised information about what a capital city means.

http://w.elnn.kr/search/

	데모	http://w.elnn.kr/
버락_오바마-미국+러시아	블라디미르/Noun_푸틴/Noun	-
버락_오바마-미국+스타워즈	아나킨/Noun_스카이워커/Noun	-
아카라카-연세대학교+고려대학교	입실렌티/Noun	입실렌티/Noun
아이폰-휴대폰+노트북	아이패드/Noun	아이패드/Noun
컴퓨터공학-자연과학+인문학	법학/Noun	게임학/Noun
플레이스테이션-소니+마이크로소프트	엑스박스/Noun_360/Number	MSX/Alpha
한국-서울+파리	프랑스/Noun	프랑스/Noun
컴퓨터-기계+인간	운영체제/Noun	일반인/Noun
게임+공부	프로그래밍/Noun	덕질/Noun
박보영-배우+가수	애프터스쿨/Noun	허각/Noun

밥+했는지	끓였/Verb	저녁밥/Noun
사랑+이별	그리움/Noun	추억/Noun
삼성-한화	노트북/Noun	후지필름/Noun
소녀시대-소녀+아줌마	아이유/Noun	에이핑크/Noun
수학-증명	경영학/Noun	이산수학/Noun
스파게티-소시지+김치	칼국수/Noun	비빔국수/Noun
아버지-남자+여자	어머니/Noun	어머니/Noun
아이유-노래+연기	송중기/Noun	송중기/Noun
안드로이드-자유	iOS/Alpha	아이폰/Noun
우주-빛	태양계/Noun_밖/Noun	NASA/Alpha
인간-직업	짐승/Noun	볼뉴르크/Noun
최현석_셰프-허세+셰프	이연/Noun_복/Noun	-
패스트푸드-체인점	영국/Noun_요리/Noun	철물/Noun

04 Learning Phrases

Learning Phrases

- ✓ To learn vector representation for phrases, this paper first find words that appear frequently together, and infrequently in other contexts.
- ✓ For example, "New York Times" and "Toronto Maple Leafs" are replaced by unique tokens in the training data, while a bigram "this is" will remain unchanged.

Method	Dimensionality	No subsampling [%]	10^{-5} subsampling [%]
NEG-5	300	24	27
NEG-15	300	27	42
HS-Huffman	300	19	**47**

Table 3: Accuracies of the Skip-gram models on the phrase analogy dataset. The models were trained on approximately one billion words from the news dataset.

	NEG-15 with 10^{-5} subsampling	HS with 10^{-5} subsampling
Vasco de Gama	Lingsugur	Italian explorer
Lake Baikal	Great Rift Valley	Aral Sea
Alan Bean	Rebbeca Naomi	moonwalker
Ionian Sea	Ruegen	Ionian Islands
chess master	chess grandmaster	Garry Kasparov

Table 4: Examples of the closest entities to the given short phrases, using two different models.

05 Additive Compositionality

Additive Compositionality

- ✓ This paper found that the Skip-gram representations exhibit another kind of linear structure that makes it possible to meaningfully combine words by an element-wise addition of their vector representations.

Czech + currency	Vietnam + capital	German + airlines	Russian + river	French + actress
koruna	Hanoi	airline Lufthansa	Moscow	Juliette Binoche
Check crown	Ho Chi Minh City	carrier Lufthansa	Volga River	Vanessa Paradis
Polish zolty	Viet Nam	flag carrier Lufthansa	upriver	Charlotte Gainsbourg
CTK	Vietnamese	Lufthansa	Russia	Cecile De

Table 5: Vector compositionality using element-wise addition. Four closest tokens to the sum of two vectors are shown, using the best Skip-gram model.

06 Comparison to Published Word Representations

Comparison to Published Word Representations

Model (training time)	Redmond	Havel	ninjutsu	graffiti	capitulate
Collobert (50d) (2 months)	conyers lubbock keene	plauen dzerzhinsky osterreich	reiki kohona karate	cheesecake gossip dioramas	abdicate accede rearm
Turian (200d) (few weeks)	McCarthy Alston Cousins	Jewell Arzu Ovitz	- - -	gunfire emotion impunity	- - -
Mnih (100d) (7 days)	Podhurst Harlang Agarwal	Pontiff Pinochet Rodionov	- - -	anaesthetics monkeys Jews	Mavericks planning hesitated
Skip-Phrase (1000d, 1 day)	Redmond Wash. Redmond Washington Microsoft	Vaclav Havel president Vaclav Havel Velvet Revolution	ninja martial arts swordsmanship	spray paint grafitti taggers	capitulation capitulated capitulating

Table 6: Examples of the closest tokens given various well known models and the Skip-gram model trained on phrases using over 30 billion training words. An empty cell means that the word was not in the vocabulary.

07 Conclusion

Conclusion

- ✓ This work show how to train distributed representations of words and phrases with the Skip-gram model and demonstrate that these representations exhibit linear structure that makes precise analogical reasoning possible.
- ✓ With the same size of datasets, word representation can be more efficiently trained.
- ✓ Hierarchical Softmax
- ✓ Negative Sampling
- ✓ Subsampling of Frequent Words

Thank you !

E-Mail : yangdonghun3@gmail.com
Phone : +82-10-2484-5894

UST UNIVERSITY OF SCIENCE & TECHNOLOGY

KISTI Korea Institute of Science and Technology Information
www.kisti.re.kr