딥러닝 세미나 Season #6

Word2Vec

2013, Google Inc.

한양대학교
컴퓨터 소프트웨어학과
인공지능 연구실
조건희

Word2Vec

Word2Vec

Word 를 Vector로 만든다!

Word(고차원 공간) → Vector(저차원 공간)

투영해준다는 의미

Word(고차원 공간) → Vector(저차원 공간)

투영해준다는 의미

Word Embedding이라고 표현하기도 함.

간단히 생각해보자

[아메리카노 카페라떼 카푸치노 마끼아또]

[	1	0	0	0]
[	아메리카노	카페라떼	카푸치노	마끼아또]

0	1	0	0
아메리카노	카페라떼	카푸치노	마끼아또

0	0	1	0
아메리카노	카페라떼	카푸치노	마끼아또

0	0	0	1
아메리카노	카페라떼	카푸치노	마끼아또

[0 0 0 1]

[아메리카노 카페라떼 카푸치노 마끼아또]

One-Hot Encoding

이렇게 간단히 해결될 문제가 아님

One-Hot Encoding의 문제점

One-Hot Encoding의 문제점

Embedding Vector에

단어의 의미가 드러나지 않음

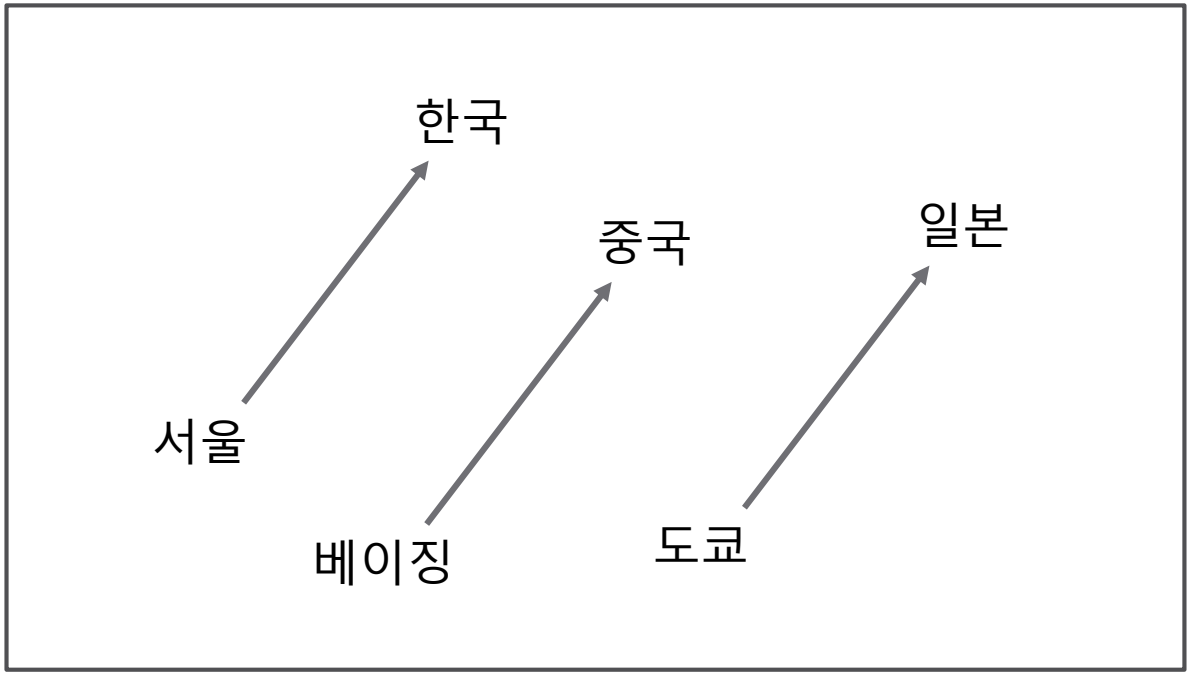
단어의 의미를 벡터화하면 뭐가 좋을까?

단어의 의미를 벡터화하면 뭐가 좋을까?

벡터 좌표가 단어의 의미를 담고 있으므로

벡터 연산을 통해 추론을 할 수 있음

http://word2vec.kr/search/



한국 – 서울 + 도쿄 = ?

Word2Vec Models

Word2Vec Models

CBOW & Skip-gram

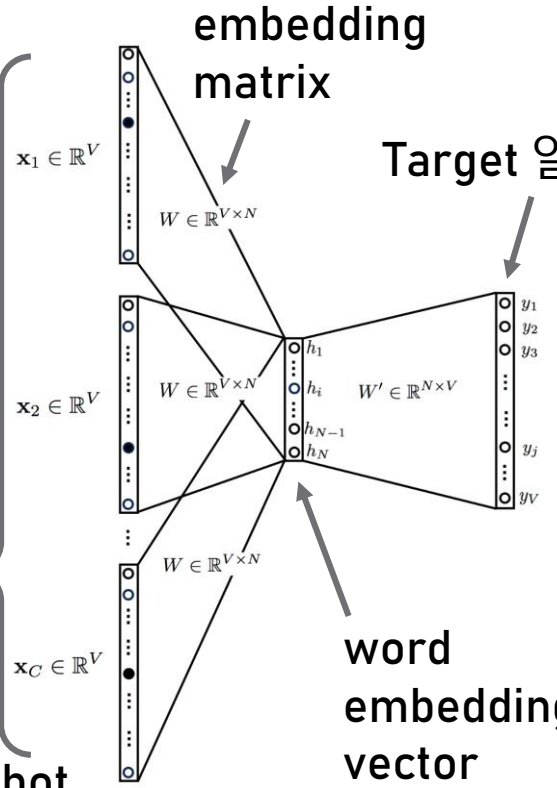
CBOW

Context(주변 단어) → Target(목표 단어) 예측

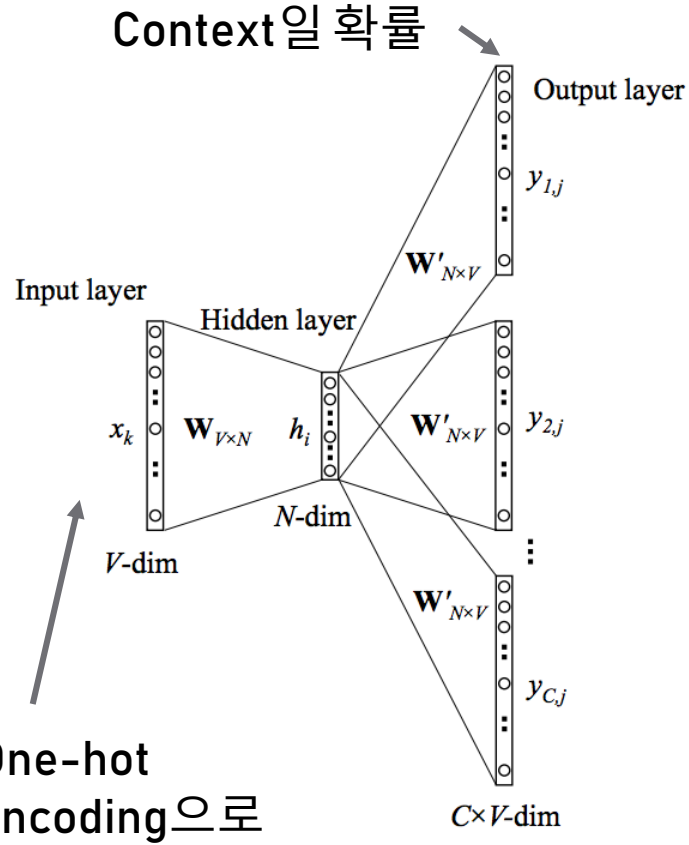
Skip-gram

Target(목표 단어) → Context(주변 단어) 예측

CBOW



Skip-gram



Skip-gram

x : One-hot encoded input word (*V*차원)

V : 단어사전 크기

W : 임베딩 매트릭스(*VxN*)

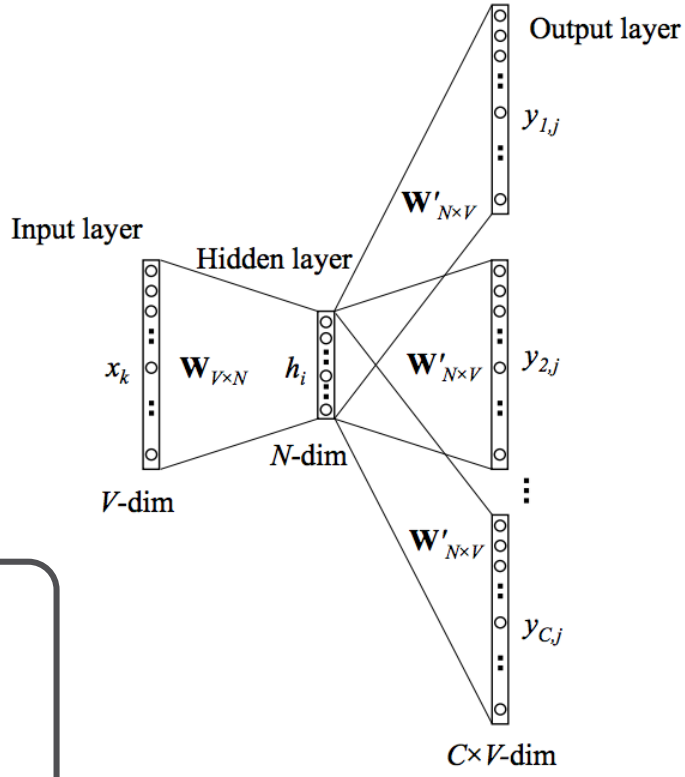
N : 임베딩 사이즈

h : word embedding 벡터 [h = x*W] (*N*차원)

W': *NxV* 매트릭스

y : output [y = softmax(h*W')] (*V*차원)

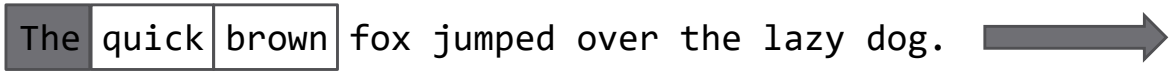
$$\underset{\mathbf{x}}{[0\ \ 0\ \ 0\ \ 1\ \ 0]} \times \underset{\mathbf{W}}{\begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix}} = \underset{\mathbf{h}}{[10\ \ 12\ \ 19]}$$



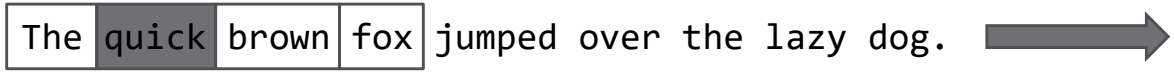
Skip-gram

말뭉치

Window size = 2
학습 데이터 샘플
(input, ouput)



(the, quick)
(the, brown)



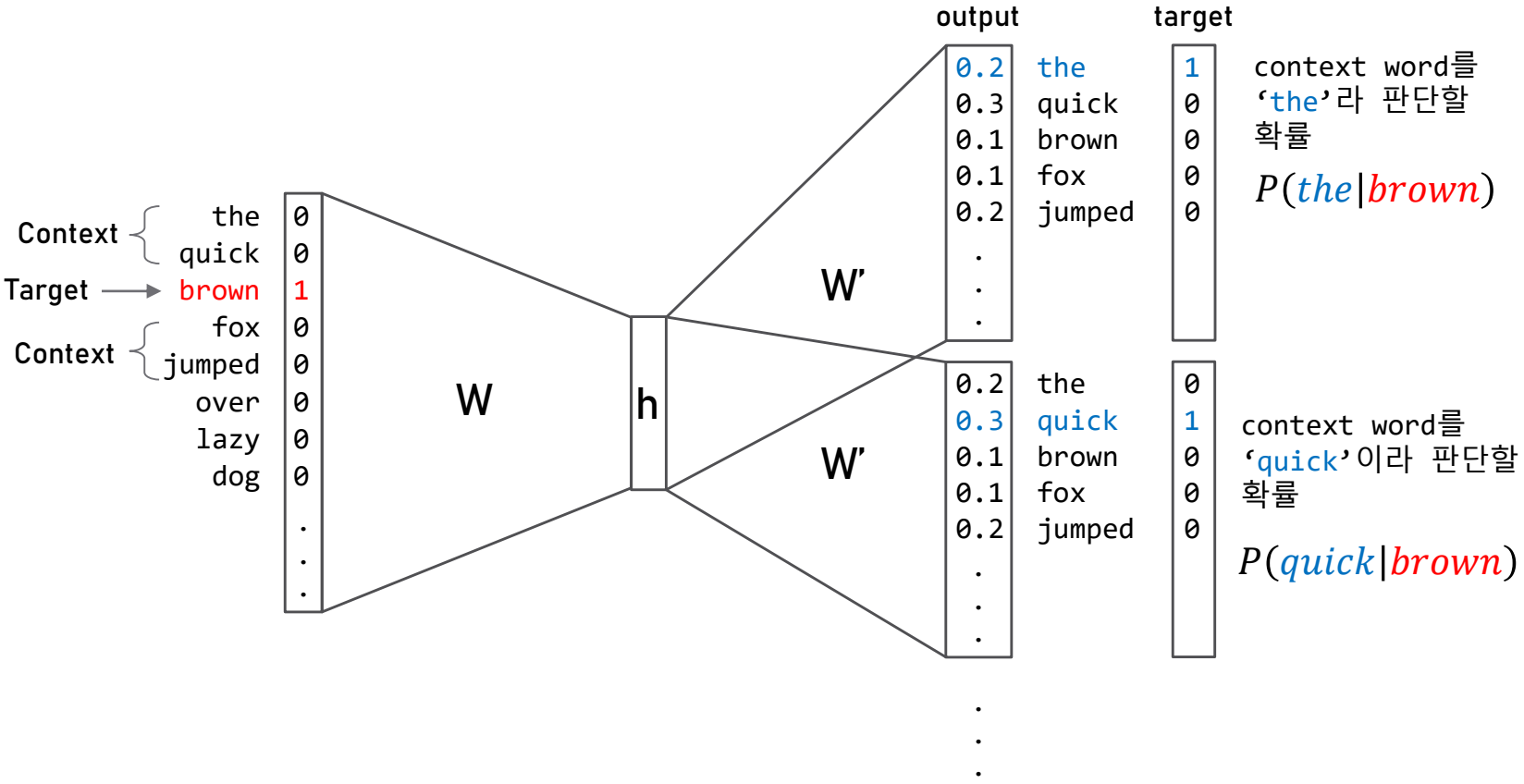
(quick, the)
(quick, brown)
(quick, fox)

(brown, the)
(brown, quick)
(brown, fox)
(brown, jumped)

말뭉치 내에 존재하는 모든 단어를 슬라이딩하며 학습데이터 샘플 구성!

Skip-gram

The quick brown fox jumped over the lazy dog.



Skip-gram

The quick brown fox jumped over the lazy dog.

$$P(quick|brown) = \frac{\exp(u_{quick}^T v_{brown})}{\sum_{w=1}^{W} \exp(u_w^T v_{brown})}$$

target 단어를 입력으로 받았을 때
이 단어가 target의 context 단어일 확률

실제 context일 경우 → 최대화
실제 context가 아닐 경우 → 최소화

Word2Vec의 학습기법

Word2Vec의 학습기법

Subsampling frequent words

Negative sampling

Subsampling frequent words

보통 말뭉치의 차원수는 10만개(V) 안팎

임베딩 차원수가 100차원(N)이라고 해도

W → 10만x100 (VxN)

W' → 100x10만 (NxV)

총 2000만개의 파라미터(free parameter)

Subsampling frequent words

보통 말뭉치의 차원수는 10만개(V) 안팎

임베딩 차원수가 100차원(N)이라고 해도

W → 10만x100 (VxN)

W' → 100x10만 (NxV)

총 2000만개의 파라미터(free parameter)

연산량이 너무 많음!

Subsampling frequent words

[샘플링을 통해 연산량을 줄이는 방법]

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

$P(w_i)$: w_i(i번째 단어)를 학습에서 제외할 확률

$f(w_i)$: w_i(i번째 단어)가 말뭉치에 등장한 비율

t: $threshold$

즉, 말뭉치에 **많이 등장한 단어**는 학습에서 **제외될 확률**이 높다!

반대로 말뭉치에 **거의 없는 단어**는 학습에 **참여할 확률**이 높다!

Negative sampling

softmax를 계산할 때

전체 단어에 대하여 계산하면

역시 연산량이 너무 많아짐.

Negative sampling

softmax를 계산할 때
전체 단어에 대하여 계산하면
역시 연산량이 너무 많아짐.

일부만 sampling해서 softmax를 계산해보자!

Negative sampling

The quick brown fox jumped over the lazy dog.

말뭉치에 존재하는 pair
(Positive sample)

(the, quick)
(the, brown)

(quick, the)
(quick, brown)
(quick, fox)

(brown, the)
(brown, quick)
(brown, fox)
(brown, jumped)

(...)

말뭉치에 존재하지 않는 pair
(Negative sample)

(the, fox)

(quick, dog)
(quick, lazy)

(brown, over)

(...)

Negative sampling

윈도우 사이즈가 2일 경우

Positive sample 4개와 Negative sample 20개

총 24개의 sample 에 대한

softmax 계산

Negative sampling

윈도우 사이즈가 2일 경우

Positive sample 4개와 Negative sample 20개

총 24개의 sample 에 대한

softmax 계산

Negative sampling

윈도우에 등장하지 않은 단어가 **Negative sample**로 등장할 확률

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

$f(w_i)$: w_i(i번째 단어)가 말뭉치에 등장한 비율

말뭉치에 등장한 비율이 높을 수록

Negative sample 로 뽑힐 확률이 높다!

Negative sampling

윈도우 사이즈가 2일 경우

Positive sample 4개와 Negative sample 20개

총 24개의 sample 에 대한

softmax 계산

Negative sampling

‹ softmax 계산 예시 ›

The quick brown fox jumped over the lazy dog.
[Context] [Target] [¬Context]

Positive sample
(brown, quick)

$$P(quick|brown)$$

$$= \frac{\exp(u_{quick}^{T} v_{brown})}{\sum_{w=1}^{24} \exp(u_{w}^{T} v_{brown})}$$

→ 높아지도록 학습 진행

Negative sample
(brown, over)

$$P(over|brown)$$

$$= \frac{\exp(u_{over}^{T} v_{brown})}{\sum_{w=1}^{24} \exp(u_{w}^{T} v_{brown})}$$

→ 낮아지도록 학습 진행

Negative sampling - NCE loss

관련 자료는 추후 추가하도록 하겠습니다 - 2019.04.15

감사합니다