

Week3. NLP

models 3-1. Word Embedding

신승진

Contents

1. Sparse Word Embedding
2. Skip-gram

3. CBOW

4. Word2vec

5. GloVe

6. fastText

7. Reference

Goal

❖ 💧 서로 다른 word embedding 모델을 이해하고, 특히 word2vec 학습 방식을 설명할 수 있다.

1. Sparse

Embedding “단어를

벡터로 만들자”

- One-hot encoding

- 예) 커피 = [1 0 0 0]
최소 = [0 1 0 0]
3 = [0 0 1 0]
잔 = [0 0 0 1]

- 단점

- 고차원 (차원의 수 == vocab 사이즈)
 - 차원의 저주(curse of dimensionality)
- 단어 확장성 ↓
- 단어의 의미를 표현하지 못함
 - 모든 단어의 거리가 동일

```
x = [
    [1,0,0,0],
    [0,1,0,0],
    [0,0,1,0],
    [0,0,0,1],
]
print(euclidean(x[0],x[1]))
print(euclidean(x[0],x[2]))
print(euclidean(x[0],x[3]))|
```

1.4142135623730951
 1.4142135623730951
 1.4142135623730951
 1.4142135623730951

- 예) 유클리디안 거리 (l2 distance)

2.



Skip-gram

“나를 보고 주변 단어를 맞춰봐”

- [참고]

- context = 주변 단어
- target = 중심 단어
- window_size = (한 방향의) 주변 단어 개수
 - context 개수 = $window_size * 2$
- 특정 단어(target)가 입력 됐을 때 그 주변 단어(context)를 맞추는 문제
- 주변 단어의 순서는 고려하지 않음
- 장점: 빈도 낮은 단어의 의미도 잘 잡아냄. 학습 데이터가 적을 때도 잘 학습

“하루 커피 최소 3잔은 마셔야 살 수 있어 _ context: “커피”, “최소”,
.” - window_size = 2
_ target: “3잔은”

Hidden Layer Weight =
“마셔야”, “살”
학습 데이터 셋 (3잔은, 커피) (3잔은,

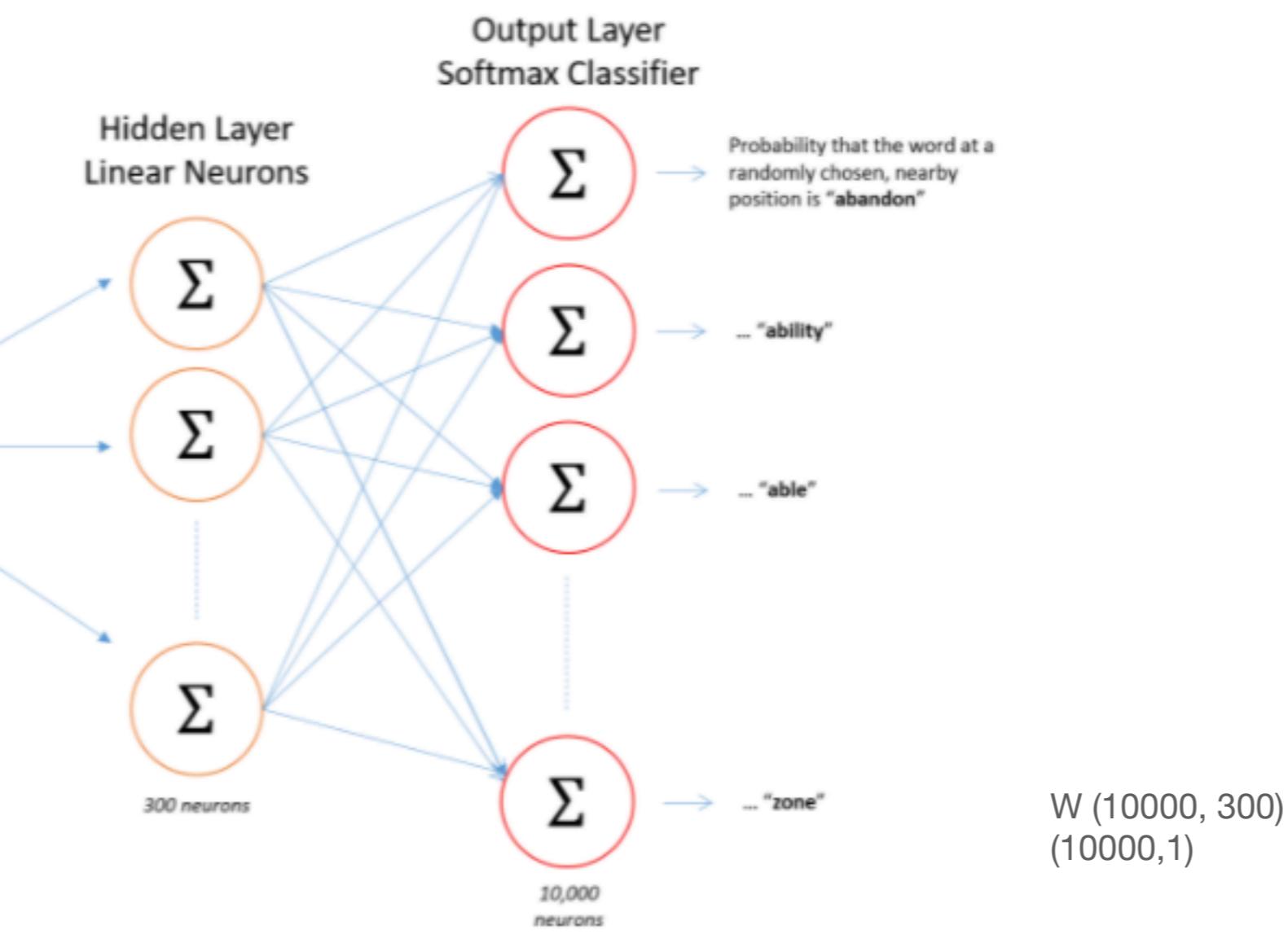


데이터가 적을 때 도 잘 학습
최소) (3잔은, 마셔야) (3잔은, 살)
Word Vector
3잔은 커피

2.

Skip-gram

“나를 보고 주변
나를 보고 주변”



(300,1)

(10000,1)

→Word Embedding W (300, 10000)

3. CBOW

“주변 단어를 보고 나를

맞춰봐” • 주변 단어(context)를 보고 특정

단어(target)를 예측

- 주변 단어의 순서는 고려하지 않음
- 장점: 빈도 높은 단어 의미 더 정확. 학습 속도 빠름

“살 수 있어.” - window_size =

2

- target: “3잔은”

- context: “커피”, “최소”,

“마셔야”, “살”

학습 데이터 셋 (커피, 3잔은)

(최소, 3잔은) (마셔야, 3잔은)

(살, 3잔은)

“하루 커피 최소 3잔은 마셔야

4. Word2vec

“간단한 신경 모델로 단어 **embedding**

학습하자”. Skip-gram 모델의 단점

- weight의 크기가 큼 (이전 예제에서만 $300 \times 10000 \times 2 = 6\text{백만}$)
- 데이터 셋이 큼
- 따라서 학습 시간 오래 걸림

- 제안1. Subsampling Frequent Words

- 자주 등장하는 단어를 코퍼스에서 제거함. 따라서 학습 데이터셋에 단어 등장 빈도를 줄임.

- 제안2. Negative Sampling (SKip-gram Negative Sampling; SKNS)

- Logistic Regression task로 변형 (softmax -> sigmoid) + negative sampling (노이즈 추가)해 복잡도 감소

Input	Output	target
3잔은	커피	1
3잔은	최소	1
3잔은	마셔야	1
3잔은	살	1

Input	Output	target
3잔은	커피	1
3잔은	주스	0
3잔은	물	0

- 학습 과정에서 Embedding Matrix, Context Matrix를 업데이트
- Embedding Matrix를 word embedding으로 사용함

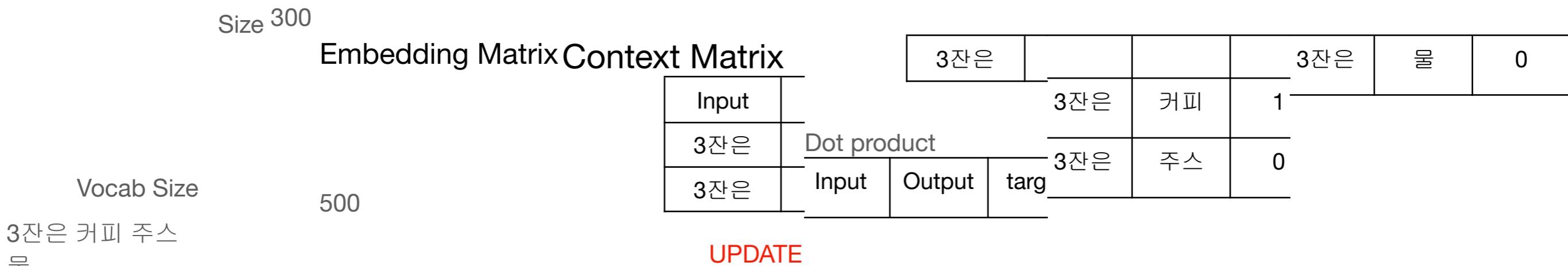
Hyperparameter

- Window size (context: (window_size-1)/2)
- # negative samples
- 데이터셋 규모 ↓ : 5~10 / 데이터셋 규모 ↑ : 2~5
- Embedding size

4. Word2vec

“간단한 신경 모델로 단어 **embedding** 학습하자”

Embedding



Embedding
Matrix
Context
Matrix

ERROR

5. GloVe

“통계 정보로 단어의 **embedding**을 만들자”

- Local context window 방식(skip-gram)은 문서 전체의 통계 정보를 반영하지 못한다는 단점이 있음. 따라서 단점을 보완하고자 전체 통계

정보를 반영하는 방법론.

- Global matrix factorization
- Latent Semantic Analysis (LSA)를 발전 시킴
- 단어가 동시에 등장하는 빈도수를 계산한 행렬(word-word co-occurrence statistics)을 생성
- $P(k|i)$ 특정 단어 i 가 등장했을 때 주변 단어 k 가 등장한 확률 계산
- 중심 단어와 주변 단어 벡터 내적 == 동시 등장 확률

n=1

“모닝 커피 좋아”

	모닝	커피	좋아
모닝	0	1	0
커피	1	0	1
조아	0	1	0

Probability and Ratio	$k = \text{solid}$	$k = \text{gas}$	$k = \text{water}$	$k = \text{fashion}$
$P(k \text{ice})$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-3}
$P(k \text{steam})$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-3}
$P(k \text{ice})/P(k \text{steam})$	8.9	8.5×10^{-2}	1.36	0.96

6. fastText

“단어 **embedding**이 아니라 **subword embedding**을 만들자”● 단어 → bag of character n-gram

- 단어 벡터를 학습하는 대신, n-gram character 벡터를 학습함 ● 단어는 n-gram character 벡터의 합으로 표현
- Skip-gram 모델을 사용
- 장점1: 학습 데이터에 등장하지 않았거나 빈도가 낮은 단어도 표현 가능 (OOV 문제 해결)
- 장점2: 접두사, 접미사 의미 학습 가능
*<ar, art, rti, tif, ifi, fic, ici,
ial, al>*

Word: artificial n=3

7. Reference

- [Illustrated Word2vec](<https://jalammar.github.io/illustrated-word2vec/>) ◆◆◆ ★★★

- [Skip-gram architecture explained](<http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/>) ♦? ♦? ♦? ★★★★
- [skip-gram&CBOW explain](<https://towardsdatascience.com/nlp-101-word2vec-skip-gram-and-cbow-93512ee24314>) ♦? ★
- [word2vec explained](<http://mccormickml.com/2017/01/11/word2vec-tutorial-part-2-negative-sampling/>) ♦? ★
- [skip-gram, CBOW 논문](<https://arxiv.org/pdf/1301.3781.pdf>) ♦? ♦? ♦? □□□
- [GloVe 한국어 설명](<https://wikidocs.net/22885>) ♦? ♦? ♦? □□□
- [word2vec 논문](<https://jalammar.github.io/illustrated-gpt2/>) ♦? ♦? ♦? □□□
- [GloVe 논문](<https://nlp.stanford.edu/pubs/glove.pdf>) ♦? □
- [fastText 논문](<https://arxiv.org/pdf/1607.04606.pdf>) ♦? □