Week11_예습과제_팽소원

Word2Vec

Word2Vec : 임베딩 모델로 단어 간의 유사성을 측정하기 위해 **분포 가설**을 기반으로 개발

분포 가설 : 같은 문맥에서 함께 자주 나타나는 단어들은 서로 유사한 의미를 가질 가능성이 높다는 가정

- 단어 간의 **동시 발생** 확률 분포를 이용해 단어 간의 유사성 측정함
- ex) '내일 자동차를 타고 부산에 간다', '내일 비행기를 타고 부산에 간다' 라는 두 문장에서 '자동차'와 '비행기'는 주변에 분포한 단어들이 유사하므로 비슷한 의미를 가질 것으로 예상
- 단어의 분산 표현(단어를 고차원 벡터 공간에 매핑하여 단어의 의미를 담는 것)을 학습함

단어 벡터화

단어를 벡터화 하는 방법은 ① 희소표현 ②밀집 표현

① 희소표현

- 원-핫 인코딩, TF-IDF 등의 빈도 기반 방법
- 대부분의 벡터 요소가 0으로 표현되어 단어 사전의 크기가 커지면 벡터의 크기도 커져 공간적 낭비 발생
- 단어 간의 유사성 반영 X, 벡터 간 유사성을 계산하는데 많은 비용 발생

② 밀집 표현 : Word2Vec

- word2vec
- 단어를 실수 벡터로 표현하기 때문에 단어 사전의 크기가 커지더라고 벡터 크기 커지지 X
- 벡터 공간상에서 단어 간의 거리를 효과적으로 계산 가능
- 밀집 표현된 벡터 = **단어 임베딩 벡터**
- word2vec은 대표적은 단어 임베딩 기법이고, 밀집 표현을 위해 CBoW와 Skip-gram이라는 방법 사용

CBoW

CBoW(Continuous Bag of Words) : 주변에 있는 단어를 가지고 중간에 있는 단어를 예측하는 방법

중심 단어 : 예측해야할 단언

주변 단어 : 예측에 사용되는 단어들

윈도 : 중심단어를 맞추기 위해 고려하는 주변 단어의 범위

슬라이딩 윈도 :윈도를 이동해가며 학습하는 방법

CBoW는 슬라이딩 윈도를 사용해 한 번의 학습으로 여러 개의 중심 단어와 그에 대한 주변 단어를 학습함

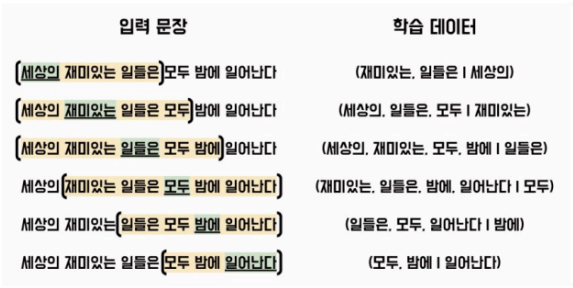


그림 6.4 CBoW의 학습 데이터 구성

다음 그림은 하나의 입력 문장에서 윈도 크기가 2일 때 학습 데이터가 어떻게 구성되는지 보여준다

학습 데이터는 (주변단어 | 중심 단어)로 구성

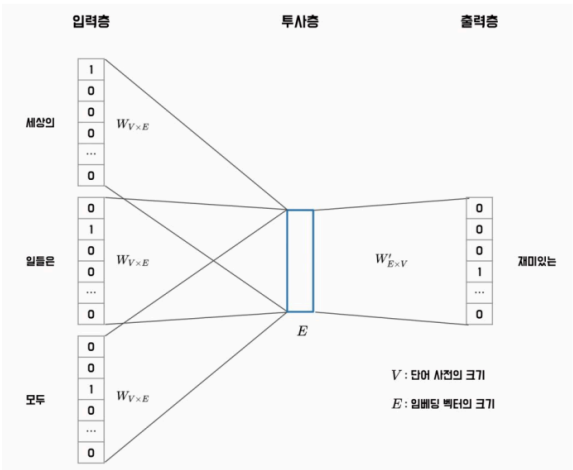


그림 6.5 CBoW 모델 구조

입력 문장 내 모든 단어의 임베딩 벡터를 평균 내어 중심 단어의 임베딩 벡터 예측

입력값은 투사층에 입력

입력값 : 입력 단어의 원-핫 벡터

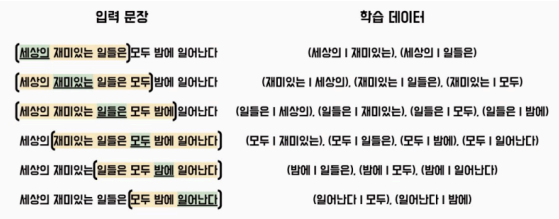
투사층: 원-핫 벡터의 인덱스에 해당하는 임베딩 벡터를 반환하는 순람표 구조

계산된 평균 벡터를 가중치 행렬 W와 곱하면 V 크기의 벡터를 얻을 수 있음

이 벡터에 소프트맥스 함수를 이용해 중심 단어 예측

Skip-gram

Skip-gram : CBoW와 반대로 중심 단어를 입력으로 받아서 주변단어를 예측하는 모델



입력 문장	학습 데이터
(세상의 재미있는 일들은)모두 밤에 일어난다	(세상의 | 재미있는), (세상의 | 일들은)
(세상의 재미있는 일들은 모두)밤에 일어난다	(재미있는 | 세상의), (재미있는 | 일들은), (재미있는 | 모두)
(세상의 재미있는 일들은 모두 밤에)일어난다	(일들은 | 세상의), (일들은 | 재미있는), (일들은 | 모두), (일들은 | 밤에)
세상의(재미있는 일들은 모두 밤에 일어난다)	(모두 | 재미있는), (모두 | 일들은), (모두 | 밤에), (모두 | 일어난다)
세상의 재미있는(일들은 모두 밤에 일어난다)	(밤에 | 일들은), (밤에 | 모두), (밤에 | 일어난다)
세상의 재미있는 일들은(모두 밤에 일어난다)	(일어난다 | 모두), (일어난다 | 밤에)

그림 6.6 Skip-gram의 학습 데이터 구성

중심 단어와 각 주변 단어를 하나의 쌍으로 하여 모델을 학습

더 많은 학습 데이터세트 추출 가능

CBoW보다 더 뛰어난 성능

드물게 등장하는 단어를 잘 학습할 수 있음

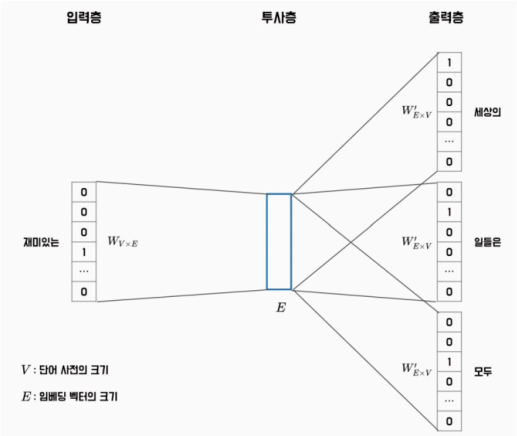


그림 6.7 Skip-gram 모델 구조

입력 단어의 원-핫 벡터를 투사층에 입력

입력 단어의 임베딩과 W 가중치와의 곱셈을 통해 V 크기의 벡터를 얻고

이 벡터에 소프트맥스 연산을 취함으로써 주변 단어 예측

계층적 소프트맥스

소프트맥스는 단어 사전이 커지면 학습속도가 느려지는 단점이 있어 이를 해결하기 위해 계층적 소프트맥스와 네거티브 샘플링 기법 적용

계층적 소프트맥스 : 출력층을 이진 트리 구조로 표현해 연산 수행

- 자주 등장하는 단어는 상위노드, 드물게 등장하는 단어는 하위 노드

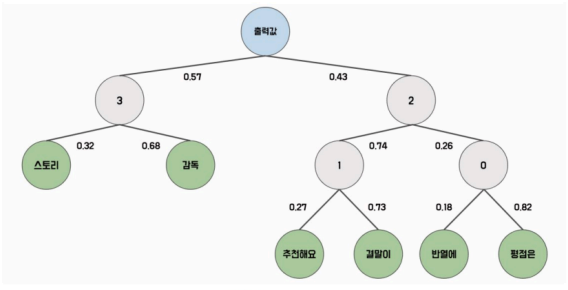


그림 6.8 계층적 소프트맥스 구조

각 노드는 학습이 가능한 벡터이고, leaf 노드는 각 단어를 의미한다

모델은 각 노드의 벡터를 최적화하여 단어 예측 수행

각 단어의 확률은 경로 노드의 확률을 곱해서 구할 수 있음

일반적인 소프트맥스 연산의 시간복잡도보다 log2배 더 작음

네거티브 샘플링

네거티브 샘플링 : Word2Vec 모델에서 사용되는 확률적인 샘플링 기법으로 전체 단어 집합에서 일부 단어를 샘플링하여 오답 단어로 사용

- 학습 윈도 내에 등장하지 않은 단어를 n개 추출하여 정답 단어와 함께 소프트맥스 연산 수행
- n = 5~20 개
- 각 단어가 추출될 확률 =

수식 6.9 네거티브 샘플링의 추출 확률

$$P(w_i) = \frac{f(w_i)^{0.75}}{\sum_{j=0}^{V} f(w_j)^{0.75}}$$

입력 데이터	출력 데이터
재미있는	세상의
재미있는	일들은
재미있는	모두

⇒

입력 데이터	출력 데이터
재미있는. 세상의	1
재미있는. 일들은	1
재미있는. 모두	1
재미있는. 걸어가다	0
재미있는. 주십시오	0
재미있는. 미리	0
재미있는. 안녕하세요	0

그림 6.9 네거티브 샘플링 모델의 훈련 데이터

실제 데이터에서 추출된 단어 쌍은 1, 네거티브 샘플링을 통해 추출된 가짜 단어쌍은 0으로 레이블

다중 분류 → 이진 분류로 학습목적 바뀜

네거티브 샘플링 모델에서는 입력 단어의 임베딩과 해당 단어가 맞는지 여부를 나타내는 레이블을 가져와 내적 연산 수행

연산을 통해 얻은 값은 시그모이드 함수를 통해 확률값으로 변환

레이블이 1인 경우는 확률이 높아지도록, 0인 경우는 확률이 낮아지도록 최적화