

# 텍스트 마이닝과 데이터 마이닝

# Part 03. 단어 임베딩과 문장 임베딩

정 정 민

## Chapter 07. 단어 임베딩, Word Embedding

---

1. 원핫 인코딩
2. 학습 기반 임베딩
3. 단어 임베딩 실습

# 원핫 인코딩

## [RECA] 원핫 인코딩 (One-hot encoding)

---

- 정의
  - 임베딩의 한 방법으로
  - 0과 1을 통해 단어를 정의
  - 특정 단어를 표현하는 위치만 1이고 나머지의 위치는 0으로 구성
- 예를 들어, 전체 단어가 'dog', 'cat', 'apple' 이 있다고 할 때,
  - dog : [1, 0, 0]
  - cat : [0, 1, 0]
  - apple : [0, 0, 1]
- 과 같이 단어의 수 만큼의 크기를 갖는 벡터가 생성
- 각 단어의 위치(1이 표현되는 위치)는 설정하기에 따라 다름
- 직관적으로 쉽게 단어를 벡터로 변환 가능

# 단어에 원핫 인코딩 적용하기

---

- 원핫 인코딩을 적용하기 위해 **문장을 단어의 형태로 분해**
  - 이를 tokenize(토큰나이즈)라고 함 - 추후 자세히 다룰 예정
  - 일단 띄어쓰기 단위로 단어 분할
- **고유한 단어 집합** 생성
  - 예를 들어, “사과는 맛있다. 바나나는 맛있다”라는 문장이 있을 때,
  - 고유한 단어는 ‘사과는’, ‘바나나는’, ‘맛있다’
- 고유 단어에 **독립된 인덱스** 부여
  - 사과는 = 0, 바나나는 = 1, 맛있다 = 2
- 벡터 생성
  - 각 단어의 인덱스에 1을 부여하고
  - 나머지 자리에 0을 채움
    - 사과는 = [1, 0, 0] / 바나나는 = [0, 1, 0] / 맛있다 = [0, 0, 1]

# 학습 기반 임베딩

## 분포 가설 (Distribution Hypothesis)


- 분포 표현 (Distribution Representation)의 이론적 기반
- 1950년대 제안된 언어학 이론으로
- “**단어의 의미**는 그 **단어가 나타나는 문맥에 의해서 결정**된다”는 아이디어를 중심으로 함
  - “우리 엄마는 물을 매일 마신다” 라는 문장이 있을 때,
    - 물 자리를 대신 할 수 있는 다른 단어들은 물과 비슷한 의미를 공유할 가능성이 큼
      - 사과 주스 / 우유 등
    - 또한, ‘물’과 ‘마신다’는 문맥적과 연관된 의미를 갖는다고 볼 수 있음
- 최신 임베딩 기법은 이러한 분포 가설을 기반으로 연구 생성 됨
  - 특정 단어의 의미를 숫자 벡터로 표현하기 위해
  - 문맥과 주변 단어의 이용해 학습 진행





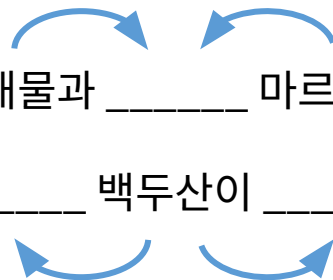
# Word2Vec

- 2013년 구글 연구진에 의해 개발된 알고리즘
- 문장 위를 움직이는 슬라이딩 윈도우 만들
- 해당 윈도우는 이동하면서 동일한 개수의 단어를 포함
- 포함된 단어들 사이의 연산을 진행해 각 단어들을 임베딩
- 두 가지 방법으로 단어를 벡터로 변환
  - CBOW : 이웃한 단어들로 가운데 단어가 무엇인지 예측하는 과정에서 임베딩을 진행
  - Skip-gram : 가운데 단어로 이웃한 단어들을 예측하는 과정으로 임베딩 진행

  
동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라 만세  
무궁화 삼천리 화려 강산대한 사람 대한으로 길이 보전하세

CBOW : 동해물과 \_\_\_\_\_ 마르고

Skip Gram : \_\_\_\_\_ 백두산이 \_\_\_\_\_



# GloVe (Global Vectors for Word Representation)

- 2014년 스탠포드 대학교 연구실 연구진에 의해 연구된 알고리즘
- 전체 글에 단어간 공동 출현 통계를 이용해
- 각 단어의 의미를 벡터로 표현
  - 각 단어 쌍이 얼마나 자주 같이 나타나는지를 기록
  - 이를 ‘공동출현행렬’ 이라고 함
- 공동으로 자주 출현하는 단어들을 벡터 공간 내 비슷한 위치에 존재하도록 임베딩

동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라 만세  
무궁화 삼천리 화려 강산대한 사람 대한으로 길이 보전하세  
...  
이 기상과 이 맘으로 충성을 다하여 괴로우나 즐거우나 나라 사랑하세  
무궁화 삼천리 화려 강산 대한 사람 대한으로 길이 보전하세



	동해물과	...	보전하세
동해물과	1		1
...			
무궁화	1		4
...			
보전하세	1	...	1

# 딥러닝을 활용한 학습 기반 단어 임베딩

---

- 그 외에도 다양한 최신 연구들이 존재
- BERT
  - 2018년 구글
  - 단어 별 중요도 기반의 모듈을 활용해
  - 문장 내적 & 외적 관계를 바탕으로 임베딩을 진행
  - 과정에서 단어 임베딩 이외에 문장 임베딩도 생성
- CLIP
  - 2021년 OpenAI
  - 이미지를 설명하는 글에서
  - 이미지와 텍스트의 공동 의미를 임베딩에 활용한 사례
- 등등

# 단어 임베딩 실습

## 원핫 인코딩 단어 적용 실습

- sklearn의 OneHotEncoder클래스 사용
- 전체 문장을 입력해야 각 단어의 벡터 생성 가능
- 또한 특정 벡터가 어떤 단어를 표현하는지 get\_feature\_names\_out() 메소드로 확인 가능

```
encoder = OneHotEncoder(sparse_output=False)
one_hot_encoded = encoder.fit_transform('사과는 맛있다 바나나는 맛있다')

one_hot_encoded
# array([[0., 0., 1.],
#        [1., 0., 0.],
#        [0., 1., 0.],
#        [1., 0., 0.]])

encoder.get_feature_names_out()
# array(['x0_맛있다', 'x0_바나나는', 'x0_사과는'], dtype=object)
```

# Word2Vec 단어 적용 실습

- Gensim 패키지를 활용
- 내장 함수로 모델 다운로드 가능
  - 시간이 다소 소요됨
- 전체 단어를 이용해 학습을 한 모델로
  - 학습 당시에 사용되지 않았던 단어는 이해하지 못함
- 기본 word2vec 모델은 특정 단어를 300개의 실수 값을 이용해 표현
- 이를 이용해 유사도 관련 어플리케이션 적용 가능
  - 유사도 계산 (Cos similarity)
  - 가장 유사한 단어 찾기



```
def use_word2vec(word):  
    model = load('word2vec-google-news-300') # 시간 소요  
    word_vector = model[word]  
    return word_vector
```



```
from scipy.spatial.distance import cosine  
  
# 두 단어 사이의 유사도  
vector1 = model[word1]  
vector2 = model[word2]  
similarity = 1 - cosine(vector1, vector2)  
  
# 가장 유사한 상위 topn개 단어 도출  
similar_words = model.most_similar(word, topn=topn)
```

## GloVe 단어 적용 실습

---

- 역시 Gensim 패키지 활용
- Word2Vec과 비슷한 사용법

```
def use_glove(word):  
    model = load('glove-wiki-gigaword-300')  
    word_vector = model[word]  
    return word_vector
```

**E.O.D**