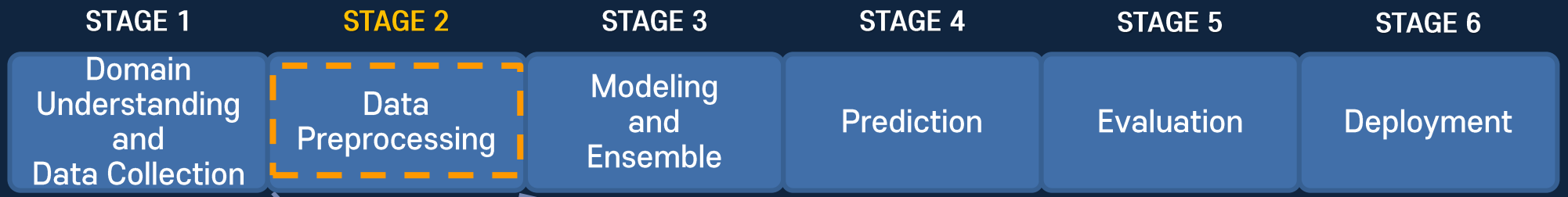
ML SESSION

#7
범주형 데이터와 Word Embedding

DATA&ANALYTICS

INDEX

1 Handling categoric data



[심화 과정]

Feature의 특징마다 다르게 처리하자

Feature { Numeric Feature
Categoric Feature

1 Handling categoric data

Data의 형태

- 범주형: 몇 가지의 범주로 나뉘어진 데이터
  - 명목형: 단순히 분류된 자료 — EX) 나라 이름, 혈액형
  - 순서형: 개개의 값들이 이산적이며 그 사이에 순서관계가 존재하는 자료 — EX) 리뷰자료 좋음 ~ 나쁨 (5 ~ 1)

- 수치형
  - 이산형: 이산적인 값을 갖는 데이터 — EX) 출산 횟수, 가구 총원
  - 연속형: 연속적인 값을 갖는 데이터 — EX) 신장, 체중

숫자로 되어 있다고 해서 무조건 수치형 데이터는 아님

1 Handling categoric data

<타이타닉 데이터>

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

범주형 변수 다루기

sklearn 패키지는 문자를 입력 값으로 처리하지 않기 때문에 숫자형으로 변환 필요

해당 Feature들을 사용하고 싶다면 적절한 전처리가 필요할 것

2 Encoding

Encoding이란?

범주형 변수를 숫자 형태로 변환하는 과정

One Hot Encoding

- N개의 클래스를 N차원의 One-Hot 벡터로 변환
- 고유 값들을 피쳐로 만들고 정답에 해당하는 열만 1로 나머지는 0으로 표시

<원본 데이터>

	Temperature	Color	Target
0	Hot	Red	1
1	Cold	Yellow	1
2	Very Hot	Blue	1
3	Warm	Blue	0
4	Hot	Red	1
5	Warm	Yellow	0
6	Warm	Red	1
7	Hot	Yellow	0
8	Hot	Yellow	1
9	Cold	Yellow	1

Temperature에 대해 One-Hot-Encoding

	Color	Target	Temp_Cold	Temp_Hot	Temp_Very Hot	Temp_Warm
0	Red	1	0	1	0	0
1	Yellow	1	1	0	0	0
2	Blue	1	0	0	1	0
3	Blue	0	0	0	0	1
4	Red	1	0	1	0	0
5	Yellow	0	0	0	0	1
6	Red	1	0	0	0	1
7	Yellow	0	0	1	0	0
8	Yellow	1	0	1	0	0
9	Yellow	1	1	0	0	0

2 Encoding

Label Encoding

- 범주형 변수의 값을 내림차순으로 정렬한 후
  0 부터 1 씩 증가하는 값으로 반환

- 숫자의 차이가 모델에 영향을 주느냐
  주지 않느냐에 따라 사용 여부 결정
  EX) 트리계열 모델에 적용 가능
  선형계열 모델에는 신중히 적용해야 함

<Temperature에 대해 Label Encoding>

	Temperature	Color	Target	Temp_Ordinal
0	Hot	Red	1	3
1	Cold	Yellow	1	1
2	Very Hot	Blue	1	4
3	Warm	Blue	0	2
4	Hot	Red	1	3
5	Warm	Yellow	0	2
6	Warm	Red	1	2
7	Hot	Yellow	0	3
8	Hot	Yellow	1	3
9	Cold	Yellow	1	1

Cold	1
Warm	2
Hot	3
Very Hot	4

2 Encoding

Target Encoding
(Mean Encoding)

- Target값 (y값)과 직접적으로 연관이 있는 인코딩
- 각 범주형 Feature 와 타겟 변수 사이의 평균값에 따라 결정됨
- One-Hot-Encoding처럼 차원이 많이 늘어나지 않기 때문에 데이터의 부피에 영향을 주지 않으며 빠른 학습에 효과적임

	Temperature	Color	Target	Temperature_mean_enc
0	Hot	Red	1	0.750000
1	Cold	Yellow	1	1.000000
2	Very Hot	Blue	1	1.000000
3	Warm	Blue	0	0.333333
4	Hot	Red	1	0.750000
5	Warm	Yellow	0	0.333333
6	Warm	Red	1	0.333333
7	Hot	Yellow	0	0.750000
8	Hot	Yellow	1	0.750000
9	Cold	Yellow	1	1.000000

Cold	1.0000
Warm	0.3333
Hot	0.7500
Very Hot	1.0000

2 Encoding

Target Encoding 방법

1. 변환 시키고자 하는 범주형 변수를 선택
2. 범주형 변수를 그룹화(group by)시키고, 타깃 변수에 대해 총합
   (예: “Temperature” 변수의 각 범주에 대한 1의 총합)
3. 범주형 변수를 그룹화 시키고, 타깃에 대한 빈도수를 총합
4. 2의 결과를 3으로 나누고, 훈련데이터의 본래 범주 값들에 업데이트

	Temperature	Color	Target	Temperature_mean_enc
0	Hot	Red	1	0.750000
1	Cold	Yellow	1	1.000000
2	Very Hot	Blue	1	1.000000
3	Warm	Blue	0	0.333333
4	Hot	Red	1	0.750000
5	Warm	Yellow	0	0.333333
6	Warm	Red	1	0.333333
7	Hot	Yellow	0	0.750000
8	Hot	Yellow	1	0.750000
9	Cold	Yellow	1	1.000000

Temperature: Hot
→ Hot 으로 groupby 한 뒤 Target의 총합: 3
→ Hot의 빈도수: 4
→ 3 / 4 = 0.75
→ 0.75 값으로 인코딩 된 피쳐 생성

Data Leakage에 의해 과적합이 일어날 리스크가 상당한 방법이기 때문에
이를 완화할 수 있는 방법을 디테일하게 알아본 후 사용하자
(smoothing 등…)

2 Encoding

그 외 다양한 인코딩 방법들

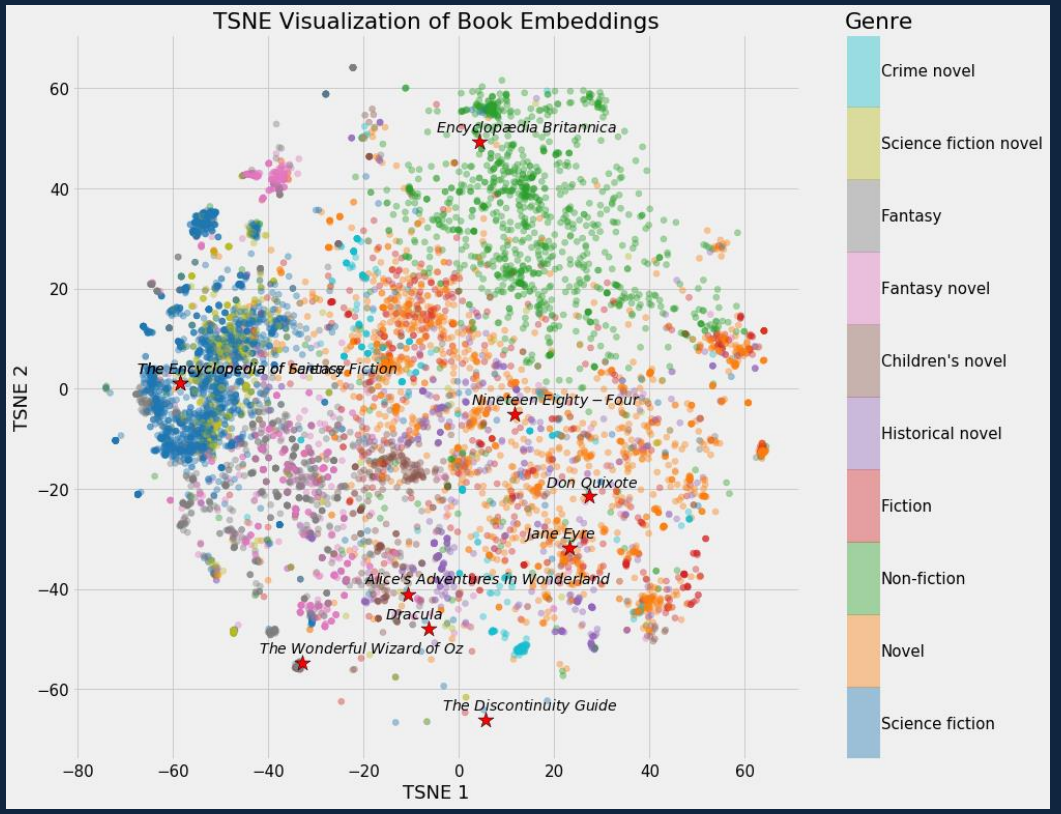
Frequency Encoding, Binary Encoding, Helmert Encoding …

https://conanmoon.medium.com/%EB%8D%B0%EC%9D%B4%ED%84%B0%EA%B3%BC%ED%95%99-%EC%9C%A0%EB%A7%9D%EC%A3%BC%EC%9D%98-%EB%A7%A4%EC%9D%BC-%EA%B8%80%EC%93%B0%EA%B8%B0-%EC%9D%BC%EA%B3%B1%EB%B2%88%EC%A7%B8-%EC%9D%BC%EC%9A%94%EC%9D%BC-7a40e7de39d4

3 Embedding

Embedding 이란?

텍스트로 되어있는 data를 숫자 벡터로 변환하는 것



3 Embedding

Sparse Representation vs Dense Representation

동물	One1	One2	one3	one4	one5	one6
고양이	1	0	0	0	0	0
강아지	0	0	1	0	0	0
말	0	1	0	0	0	0
토끼	0	0	0	1	0	0
원숭이	0	0	0	0	0	1
호랑이	0	0	0	0	1	0

VS

동물	W2v_1	W2v_2
고양이	0.3	-0.13
강아지	0.62	0.04
말	0.17	0.32
토끼	-0.21	-0.08
원숭이	0.62	0.29
호랑이	-0.42	0.73

<Sparse data>

- 벡터나 행렬 값 중 대부분이 0이고 몇몇만 값이 존재
- 단어 간의 관계를 전혀 반영하지 못함
- 차원이 커서 시간도 오래걸림

<Dense data>

- 모든 차원이 값을 갖는 벡터로 표현
- 단어 간의 관계를 반영할 수 있음
- 차원의 수를 내가 직접 결정할 수 있음

3 Embedding

Dense Representation의 장점

1. 적은 차원으로 대상 표현 가능

   → 차원의 저주를 피해 모델의 학습력을 높일 수 있음

   → Dense representation으로 표현할 때에는 20 ~ 300차원 정도로 표현함

2. 더 큰 일반화 능력을 가짐

   학습 데이터 셋에서 강아지 빈도가 높고 멍멍이 빈도는 낮았다고 가정

   Sparse의 경우 강아지에 대해 잘 알더라도 멍멍이를 알게 되는 것은 아님

   하지만 Dense의 경우에 강아지와 멍멍이가 비슷한 벡터로 표현이 된다면?

   → 강아지에 대한 정보가 멍멍이에도 일반화 될 수 있음

4 Word2Vector

Dense representation 장점의 전제: 단어 임베딩이 잘 되어 있어야 함!!

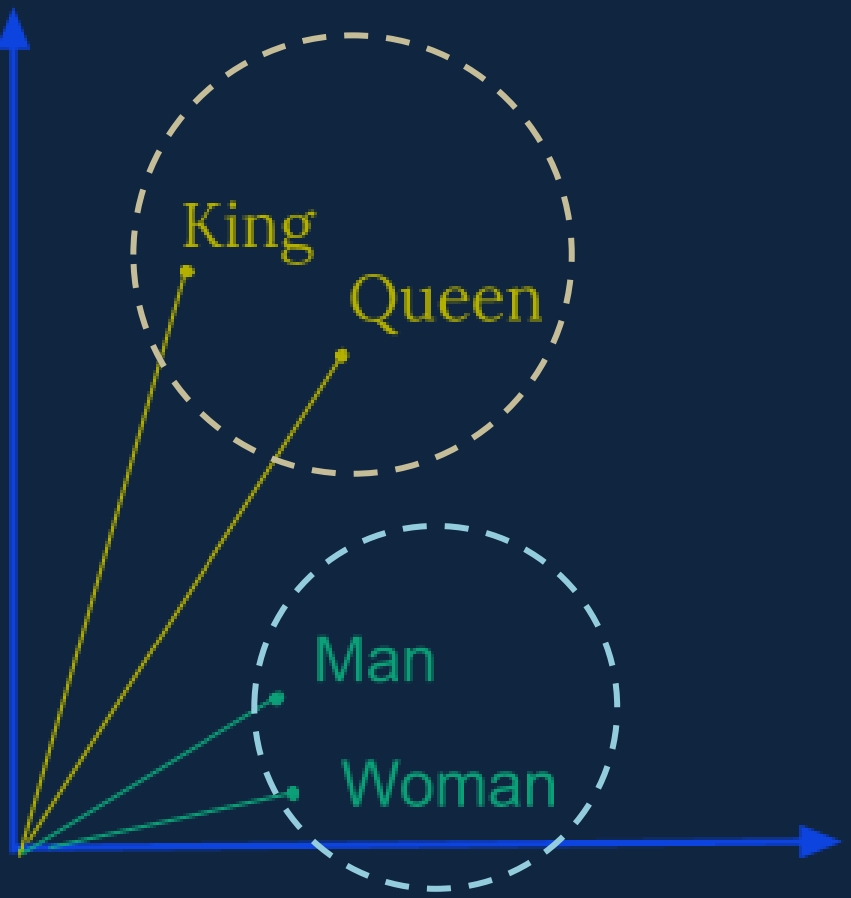
W2V

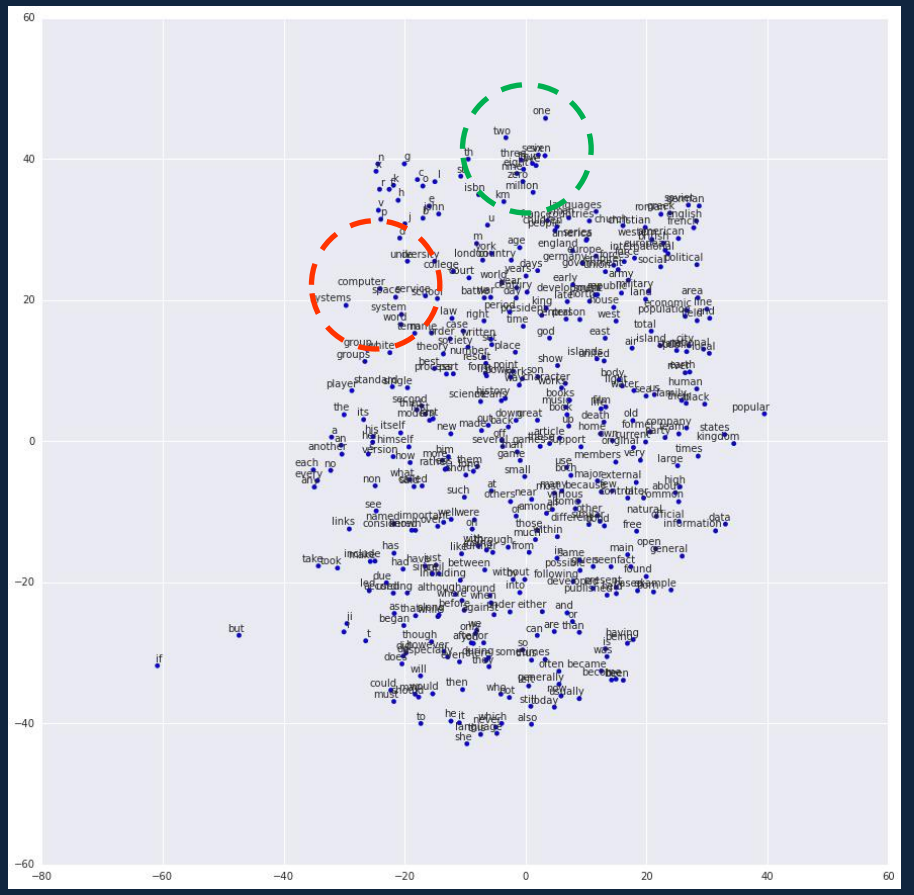
- 단어 임베딩을 학습하는 방법: Word2Vector
- Word2Vec의 기본개념: 단어의 주변을 보면 그 단어를 알 수 있다.
- W2V의 방법 2가지
  - CBOW
  - skip-gram
- 신경망(Neural Net)의 개념이 나오나 자세히 다루지 않을 것.

W2V 예시

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

4 Word2Vector





5 W2V: CBOW

< 빈칸에 들어갈 말은? >

나는 주말에 _ _ _ 을(를) 하러 갈 예정이다.

5 W2V: CBOW

CBOW

- 주변 단어들을 통해 빈칸의 단어를 유추하는 방법
  = 맥락으로 타겟 단어를 예측

- 앞 뒤 어느정도 까지의 맥락을 볼 것인가?
  → Window Size 하이퍼 파라미터 활용

_ _ 주말에 파마를 하러 갈 예정이다.

나는 _ _ _ _ 파마를 하러 갈 예정이다.

나는 주말에 _ _ _ 하러 갈 예정이다.

나는 주말에 파마를 _ _ 갈 예정이다.

나는 주말에 파마를 하러 _ 예정이다.

나는 주말에 파마를 하러 갈 _ _ _ _.

window_size = 2
→ 해당 빈칸 앞 뒤 2단어까지 볼 것

5 W2V: CBOW

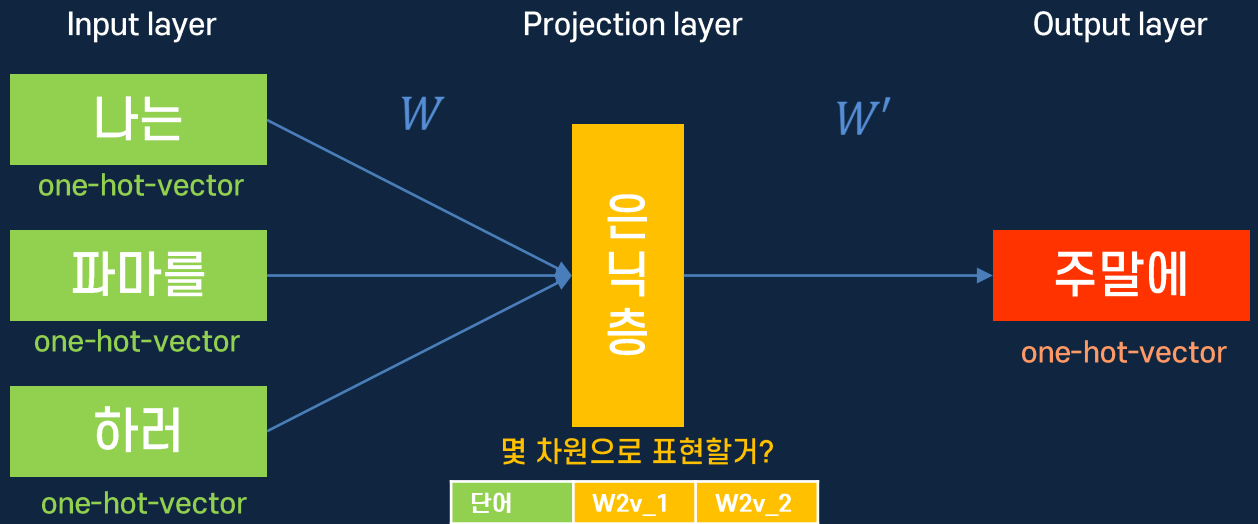
<참고>

중심 단어 / 주변 단어

중심 단어	주변 단어
The fat cat sat on the mat	[1, 0, 0, 0, 0, 0]	[0, 1, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0]
The fat cat sat on the mat	[0, 1, 0, 0, 0, 0]	[1, 0, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0], [0, 0, 0, 1, 0, 0]
The fat cat sat on the mat	[0, 0, 1, 0, 0, 0]	[1, 0, 0, 0, 0, 0], [0, 1, 0, 0, 0, 0], [0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 1, 0]
The fat cat sat on the mat	[0, 0, 0, 1, 0, 0]	[0, 1, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0], [0, 0, 0, 0, 1, 0], [0, 0, 0, 0, 0, 1]
The fat cat sat on the mat	[0, 0, 0, 0, 1, 0]	[0, 0, 1, 0, 0, 0], [0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 0, 1], [0, 0, 0, 0, 0, 0]
The fat cat sat on the mat	[0, 0, 0, 0, 0, 1]	[0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 1, 0], [0, 0, 0, 0, 0, 1]
The fat cat sat on the mat	[0, 0, 0, 0, 0, 1]	[0, 0, 0, 0, 1, 0], [0, 0, 0, 0, 0, 1]

CBOW 구조

나는 _ _ _ 파마를 하러 갈 예정이다.



몇 차원으로 표현할거?

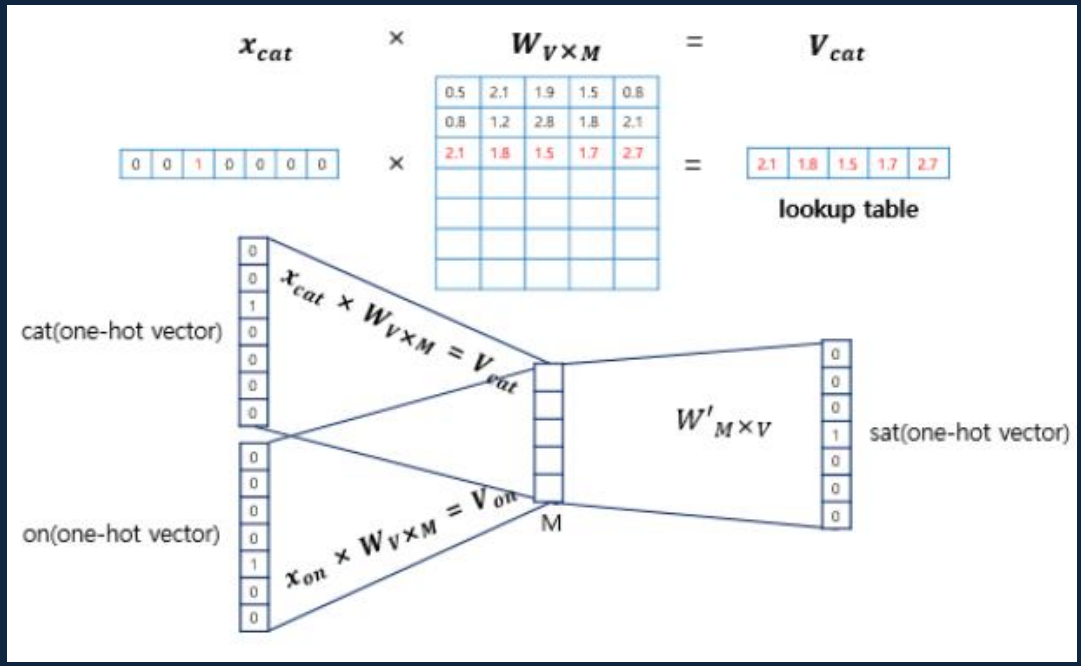
단어	W2v_1	W2v_2
나는	0.3	-0.13
주말에	0.62	0.04
파마를	0.17	0.32
하러	-0.21	-0.08
갈	0.62	0.29
예정이다	-0.42	0.73

5 W2V: CBOW

CBOW 구조

<참고>

중심 단어	주변 단어
The fat cat sat on the mat	[1, 0, 0, 0, 0, 0]	[0, 1, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0]
The fat cat sat on the mat	[0, 1, 0, 0, 0, 0]	[1, 0, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0], [0, 0, 0, 1, 0, 0]
The fat cat sat on the mat	[0, 0, 1, 0, 0, 0]	[1, 0, 0, 0, 0, 0], [0, 1, 0, 0, 0, 0], [0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 1, 0]
The fat cat sat on the mat	[0, 0, 0, 1, 0, 0]	[0, 1, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0], [0, 0, 0, 0, 1, 0], [0, 0, 0, 0, 0, 1]
The fat cat sat on the mat	[0, 0, 0, 0, 1, 0]	[0, 0, 1, 0, 0, 0], [0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 1, 0], [0, 0, 0, 0, 0, 1]
The fat cat sat on the mat	[0, 0, 0, 0, 0, 1]	[0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 1, 0], [0, 0, 0, 0, 0, 1]
The fat cat sat on the mat	[0, 0, 0, 0, 0, 1]	[0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 1, 0]



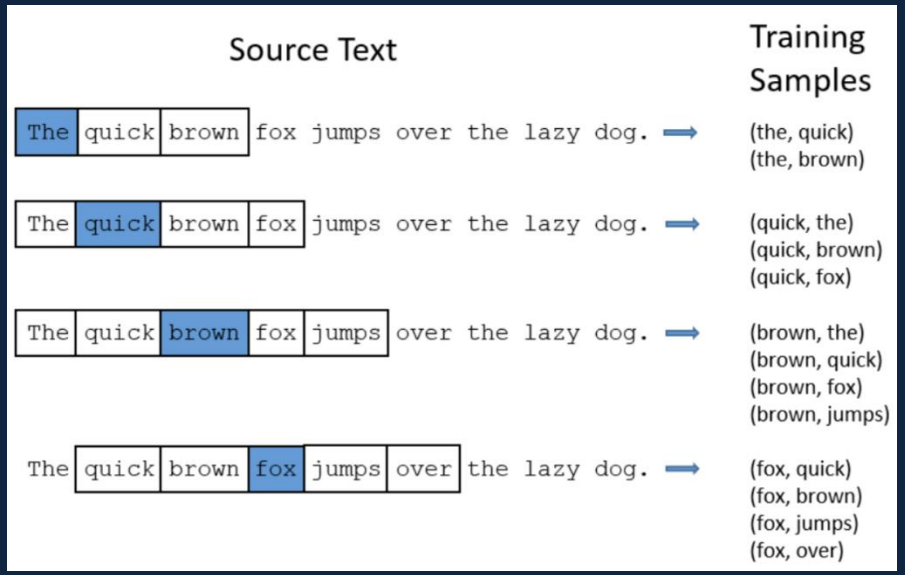
6 W2V: skip-gram

< 빈칸에 들어갈 말은? >

_ _ _ _ _ 파마를 _ _ _ _ _ _ _ _.

이번에는 한 단어로 다른 단어를 예측해야 함 !!

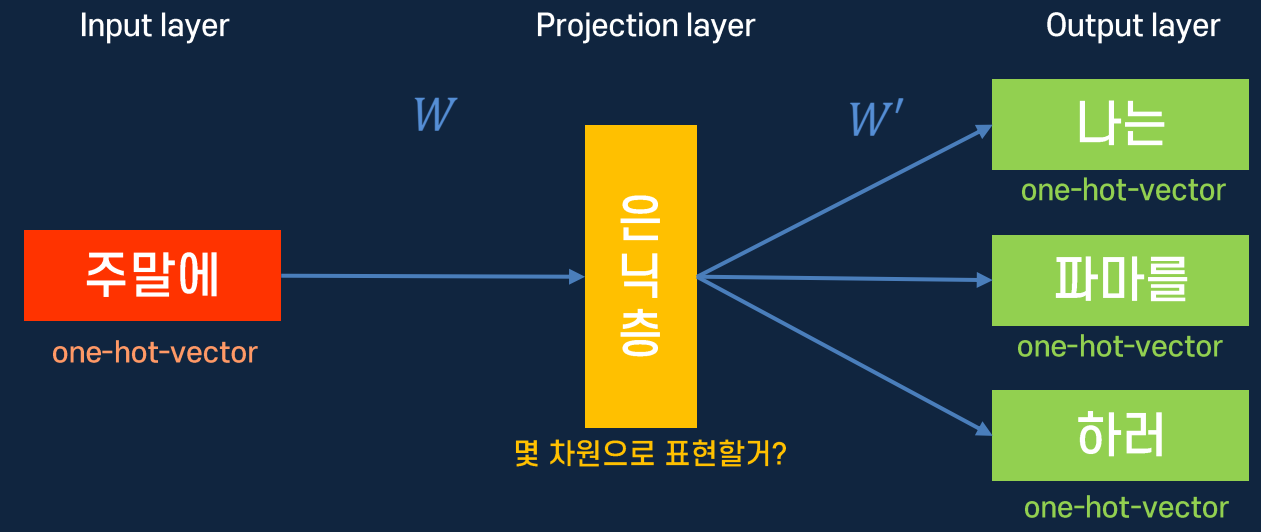
6 W2V: skip-gram



6 W2V: skip-gram

Skip-Gram 구조

- 중심 단어를 통해 주변 단어를 예측
- CBOW에 비해 중심단어의 업데이트 기회가 많다.
- 매커니즘 자체는 동일

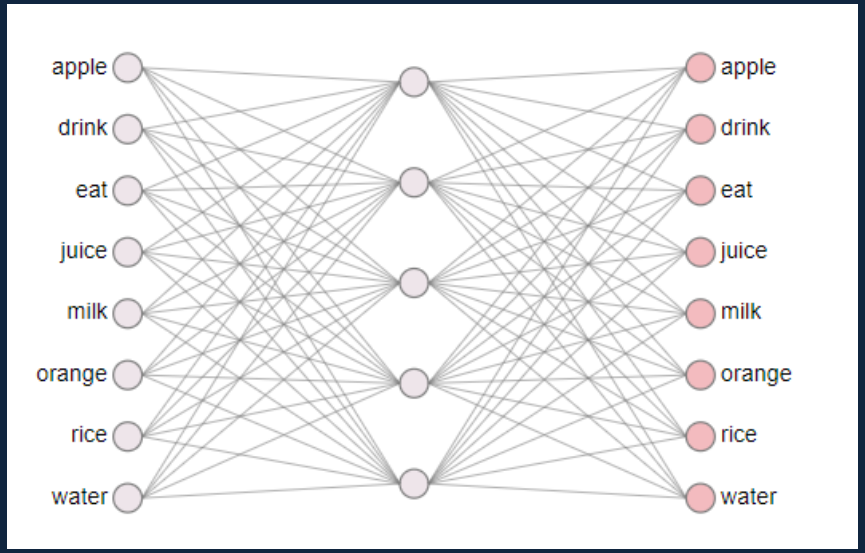


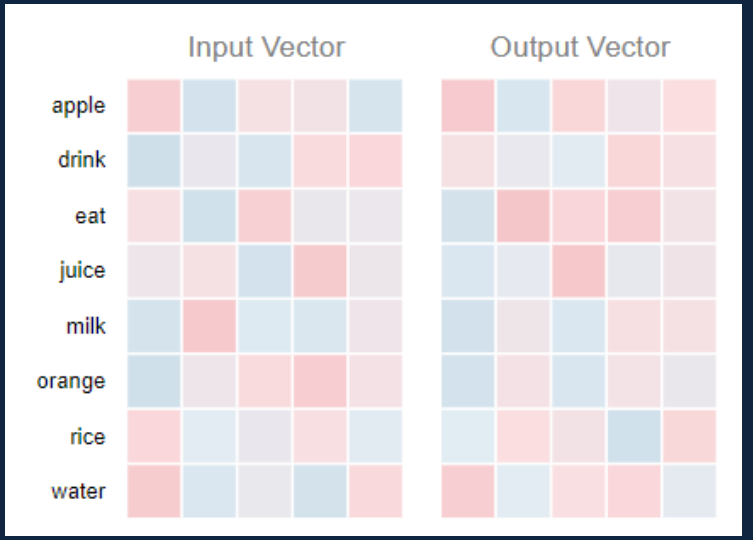
일반적으로 CBOW 방법보다 skip-gram이 성능이 더 좋다고 함

6 W2V

w2v의 구조 시각화

https://ronxin.github.io/wevi/





과제

Word2Vector를 이용해 캐글 baseline2 점수 넘어보기

Q n A

THANK YOU