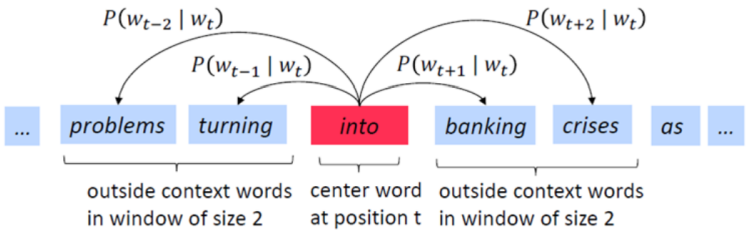
▼ *2주차 예습과제*

0.Word2Vec

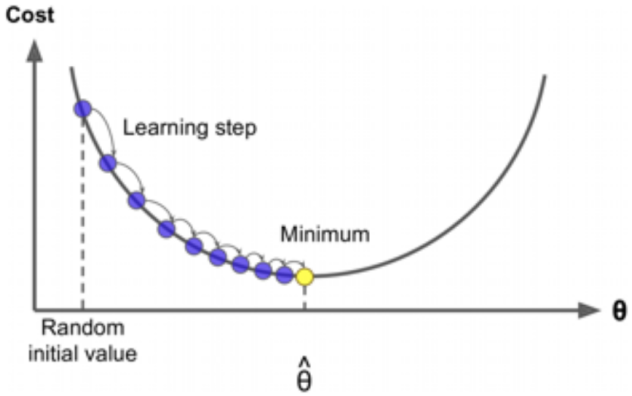
텍스트를 컴퓨터가 인식할 수 있도고 숫자 형태로 변환해야한다. 이때 벡터나 행렬로 변환한다. 대표적인 방법으로는 1)One Hot Encoding 과 2)Word Embedding 이 있다. One Hot Encoding은 0과 1로 구성된 숫자열이기에 여러 단어간의 유사성을 반영하지 못한다는 문제점이 있었다. 그렇기에 Word Embedding이 발전되게 되었다.



위 그림은 앞서 강의에서 다룬 Word2Vec의 과정을 나타낸 그림이다. center word의 양쪽 방향으로 window만큼의 단어를 context words 라고 설정할 수 있다. 그 후, One Hot Encoding 을 이용하여 center word, context words의 벡터를 생성한다. 이 벡터를 입력, 출력 벡터로 사용한다. 이때 context 벡터가 입력 벡터일 경우 CBOW 방식을 사용하고, center 벡터가 입력 벡터일 경우는 skip-gram 방식을 사용한다.

1. Optimization: Gradient Descent

우리는 단어 벡터들을 학습을 통해 최적화시킨다. word 벡터에 대한 미분을 통항 cost 함수의 경사를 계산할 수 있다. 이때 변수는 θ이고 함수의 변수 θ에서의 음의 경사가 점점 줄어드는 방향으로 향하면서 최소값을 찾을 수 있다.



2. Stochastic Gradient Descent

경사 하강법에는 하나의 값을 얻기 위해서는 긴 시간이 필요하다는 단점이 있었다. 따라서 확률적 경사 하강이라는 방법으로 발전되었다. 확률적 경사 하강법은 경사 하강법과 달리 전체 데이터가 아닌 그 중 일부 sample만 사용한다. 일부인 sample에 대해서만 기울기를 계산하기 때문에 더 빠른 계산이 가능하다. 하지만 noise가 크다는 단점이 있다.

3. CBOW, Skip-grams

CBOW(continuous bag of words)-> 모든 context word를 이용하여 center word 예측

Skip-grams->center word를 이용하여 context word 예측

4. Co-occurrence matrix

- I like deep learning.
- I like NLP.
- I enjoy flying.

counts	I	like	enjoy	deep	learning	NLP	flying	.
I	0	2	1	0	0	0	0	0
like	2	0	0	1	0	1	0	0
enjoy	1	0	0	0	0	0	1	0
deep	0	1	0	0	1	0	0	0
learning	0	0	0	1	0	0	0	1
NLP	0	1	0	0	0	0	0	1
flying	0	0	1	0	0	0	0	1
.	0	0	0	0	1	1	1	0

위 그림과 같이 window based co-occurence matrix는 특정 크기의 window 안에 나타나는 단어들을 모두 count하여 단어가 다른 단어에 인접하게 몇 번 사용되었는지 matrix로 나타낸다. 비슷한 의미를 가진 단어들은 유사한 문맥에서 사용될 가능성이 높기 때문에, 유사한 의미를 가진 단어들끼리 비슷한 단어 벡터를 가지게 된다. 하지만 이런 matrix 사용은 단어 수가 많아질 경우 지나치게 부피가 커진다는 단점이 있다.

이런 문제점을 해결하기 위해 차원을 낮추어 벡터를 설정하는 방법이 있다. 차원을 낮추기 위해 singular balue decomposition을 실시할 수 있다. 임의의 행렬을 설정하여 3개의 행렬로 분리하는 방법이다.

5. Count Based vs. Direct Prediction

1) Count Based

-LSA, HAL, COALS, Hellinger-PCA

-학습 속도가 빠르다

-통계를 효과적으로 사용 가능

-단어의 유사성을 알아내는데에만 사용된다

-큰 숫자에 불균형하게 중요도 부여

2) Direct prediction

-Skip-gram, CBOW, NNLM, HLBL, RNN

-window 내의 데이터만 사용하기에 통계 정보를 효율적으로 활용하지 않는다.

-단어 유사성뿐만 아니라 복잡한 패턴도 분석 가능하다.

6. GloVe

앞서 설명한 Direct Prediction과 Count Based 방법을 합쳐서 적용한다. 단어간 유사성을 측정하면서 전체적인 데이터 분석이 가능하게 하기 위함이다.

$$J = \sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2$$

목적함수는 위와 같이 정의 될 수 있다.

이런 방식을 통해 작은 데이터에도 효과적으로 적용가능하고 등장 빈도가 낮은 단어도 효과적인 word vector를 생성할 수 있다.

7. Word vector Evaluation

1) 본질적인 평가

:서브 작업에 대한 평가

-빠른 계산

-시스템을 이해하는데 도움이 되지만 실제 작업에 대한 상관관계가 정해지지 않으면 도움이 되는지 확실하지 않다.

2)외적인 평가

:실제 작업 성능에 대한 평가

-시간이 오래 걸림

-서브작업의 문제인지 그것과 상호작용하는 다른 시스템의 문제인지 명확하게 밝히는데 어려움이 있다.

8. Word ambiguity

한 단어가 다양한 의미를 가지는 경우가 많다. 이럴 경우 하나의 벡터가 이 모든 의미를 포함하는데 어려움이 있다. 그렇기에 다음 방법을 이용하여 의미를 구분한다.

1) Multiple sensors for a word

하나의 단어가 여러개의 벡터 cluster를 형성하는 경우 그 단어를 여러개로 분류하여 벡터를 나눈다. 단어 주변의 window word를 클러스터링하여 다른 클러스터들에 벡터를 다시 부여하는 방식으로 분류한다.

2) Weighted average

여러 의미를 가지고 있는 한 단어의 각 의미 벡터들에 가중치를 부여하여 weighted average를 도출하는 방법