

딥러닝 세미나 2주차
김동협

word2vec

DSP Lab, Inha University

August , 2021



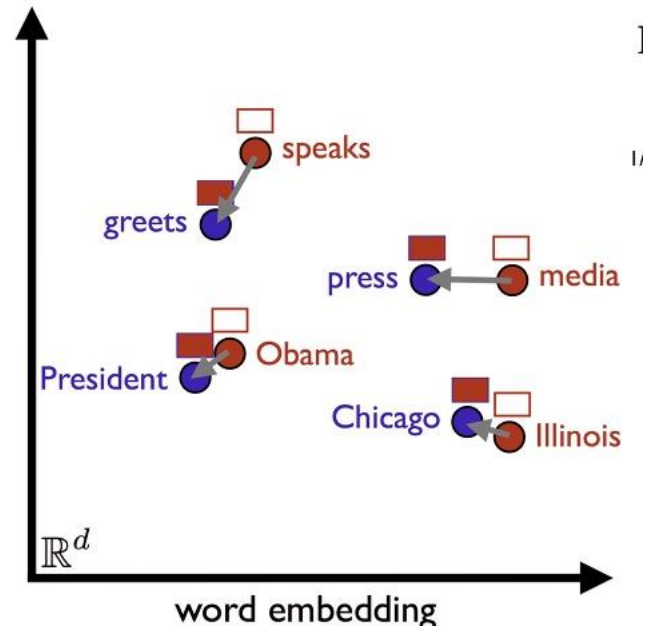
-word2vec

- 하나의 은닉층만 존재. -> Shallow Neural Network



-word2vec

- 목표 : 단어의 분산 표현을 얻는 것 =>
비슷한 단어는 비슷한 위치에 놓이게 학습시키는 것



구글에서 개발한 Word2Vec은 분포 가설(distributional hypothesis)을 가정 하에 표현한 분산 표현을 따릅니다. 예를 들어 '강아지'라는 단어는 '귀엽다', '예쁘다', '애교' 등의 단어와 같이 자주 등장한다고 해봅시다. 그에 따라 분포 가설에 맞춰 해당 단어들을 벡터화한다면 유사한 값이 나올 것입니다. 즉, 의미적으로 가까운 단어가 된다는 뜻입니다.



-word2vec

- CBOW -> 이 그림을 구현하는 것이 목표

그림 3-2 주변 단어들을 맥락으로 사용해 “?”에 들어갈 단어를 추측한다.

you ? goodbye and I say hello.



$\begin{pmatrix} \text{you} \\ \text{goodbye} \end{pmatrix}$
맥락

모델

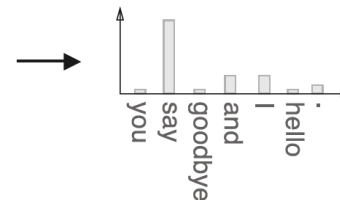


그림 3-3 추론 기반 기법: 맥락을 입력하면 모델은 각 단어의 출현 확률을 출력한다.

- 문장을 받아서 단어를 벡터화시킨다.

그림 3-4 단어, 단어 ID, 원핫 표현

단어(텍스트)	단어 ID	원핫 표현
$\begin{pmatrix} \text{you} \\ \text{goodbye} \end{pmatrix}$	$\begin{pmatrix} 0 \\ 2 \end{pmatrix}$	$\begin{pmatrix} (1, 0, 0, 0, 0, 0, 0) \\ (0, 0, 1, 0, 0, 0, 0) \end{pmatrix}$



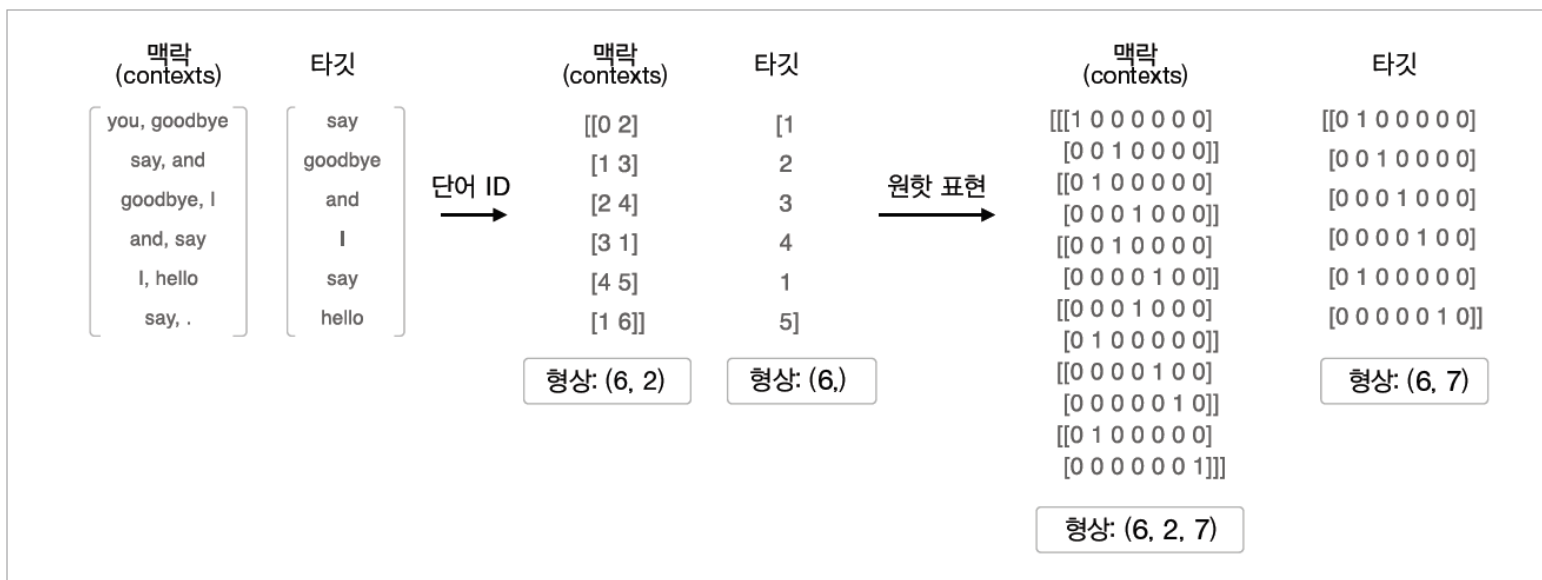
-word2vec

- 문장을 받아서 단어를 벡터화시킨다.

그림 3-4 단어, 단어 ID, 원핫 표현

단어(텍스트)	단어 ID	원핫 표현
$\begin{pmatrix} \text{you} \\ \text{goodbye} \end{pmatrix}$	$\begin{pmatrix} 0 \\ 2 \end{pmatrix}$	$\begin{pmatrix} (1, 0, 0, 0, 0, 0, 0) \\ (0, 0, 1, 0, 0, 0, 0) \end{pmatrix}$

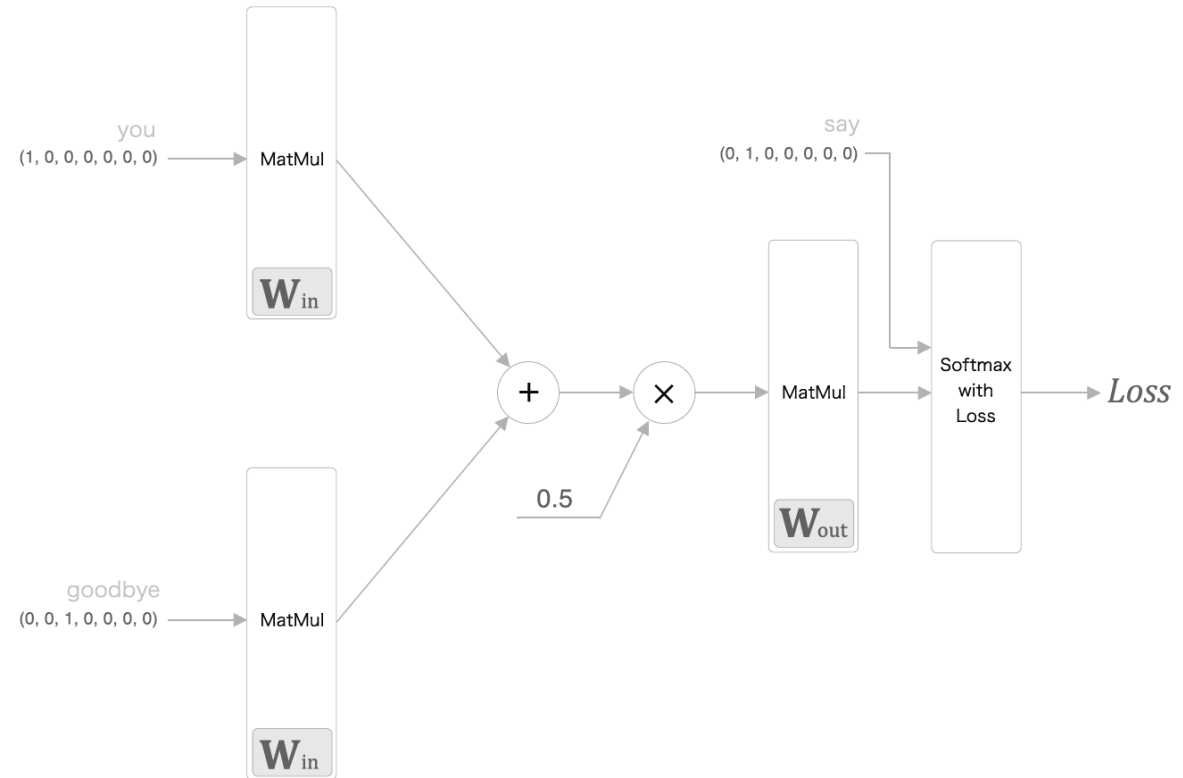
그림 3-18 '맥락'과 '타깃'을 원핫 표현으로 변환하는 예



-word2vec (순전파)

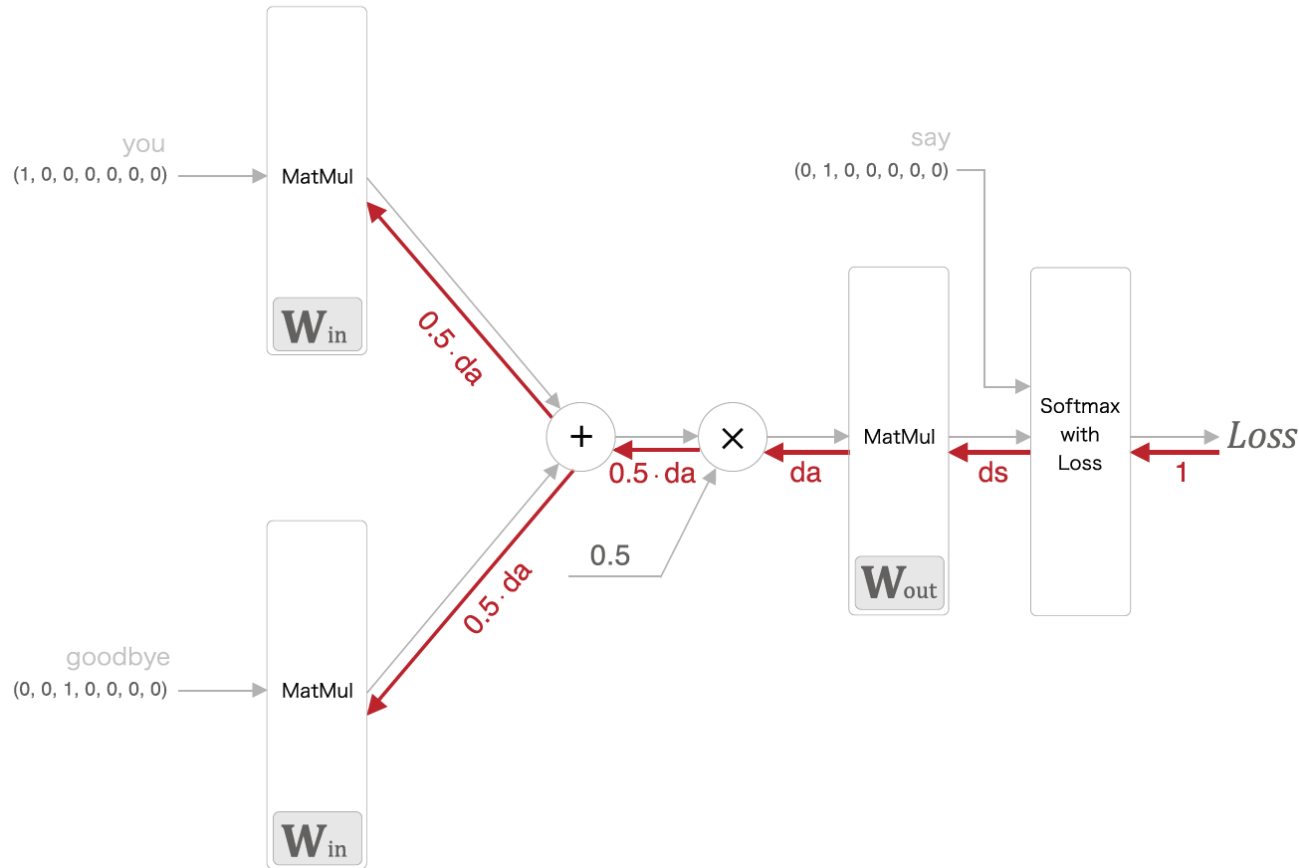
- 문장을 받아서 one-hot벡터화시킨다.
- (CBOW기준) 타깃 주변 단어(맥락)을 준비한다
- 벡터를 은닉층에 (행렬곱)통과시킨다 (W_{in} , W_{out})
- Softmax 후 타깃 레이블과 비교한다
- Cross Entropy 오차 평가

그림 3-19 CBOW 모델의 신경망 구성



-word2vec (역전파)

그림 3-20 CBOW 모델의 역전파(역전파의 흐름은 두꺼운(붉은) 화살표로 표시)



$Ds=y-t$

```
def backward(self, dout=1):  
    ds = self.loss_layer.backward(dout)  
    da = self.out_layer.backward(ds)  
    da *= 0.5  
    self.in_layer1.backward(da)  
    self.in_layer0.backward(da)  
    return None
```

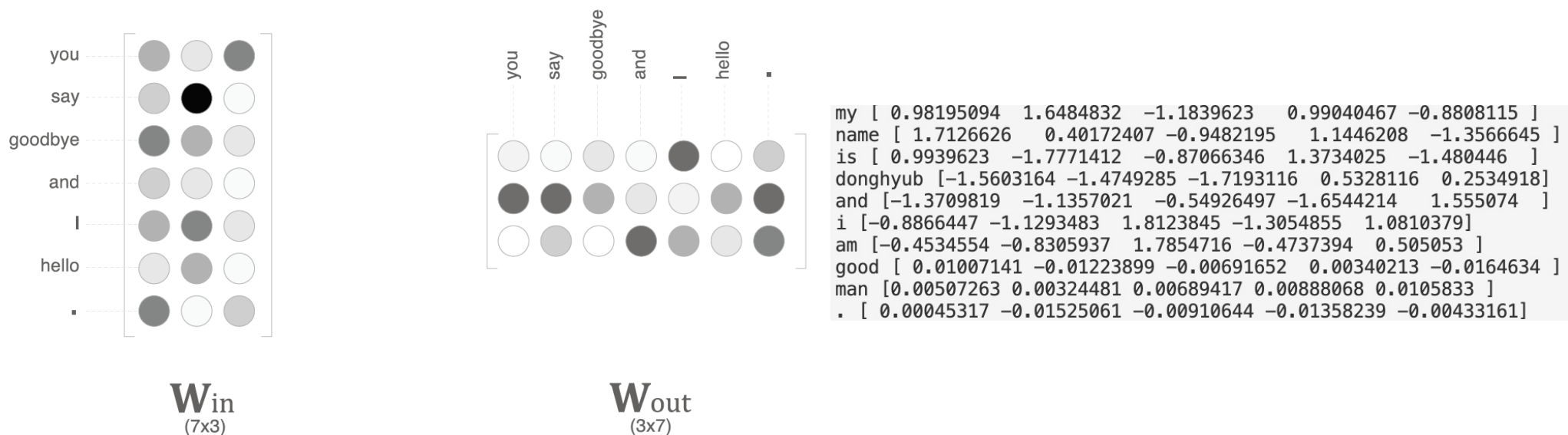
```
class MatMul:  
    def backward(self, dout):  
        W, = self.params  
        dx = np.dot(dout, W.T)  
        dW = np.dot(self.x.T, dout)  
        self.grads[0][...] = dW  
        return dx
```



-word2vec

- 이렇게 학습시켜 나온 W_{in} 이 우리가 찾는 단어의 분산

그림 3-15 각 단어의 분산 표현은 입력 측과 출력 측 모두의 가중치에서 확인할 수 있다.



-skip-gram 모델

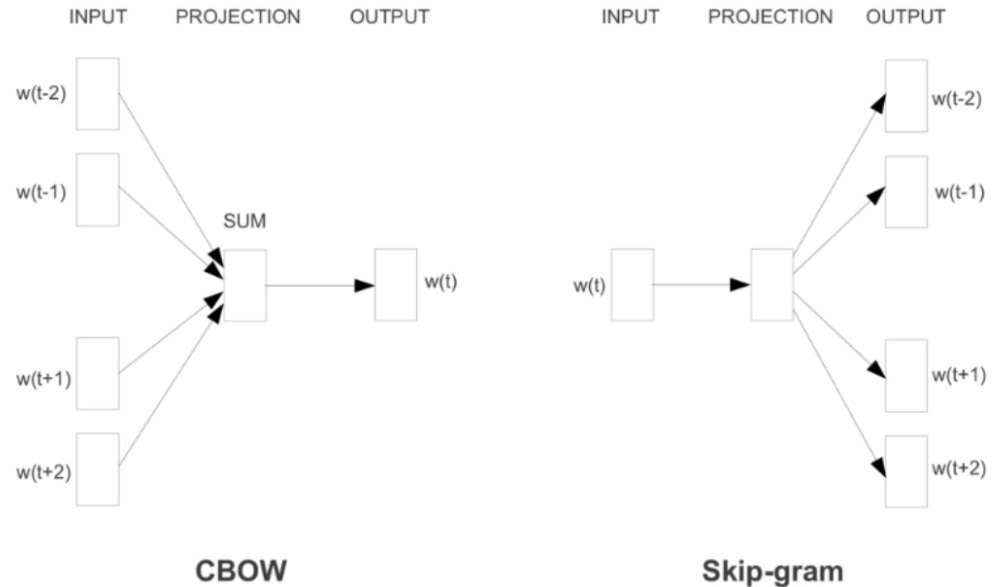
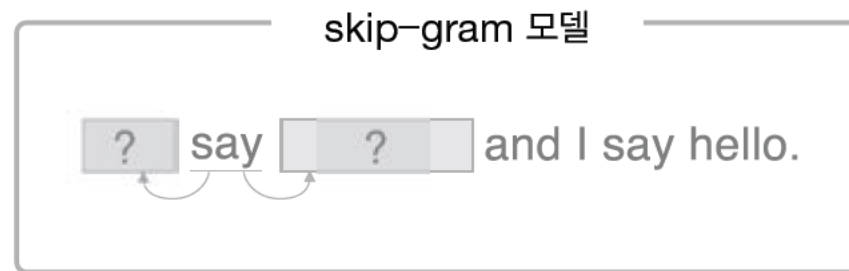
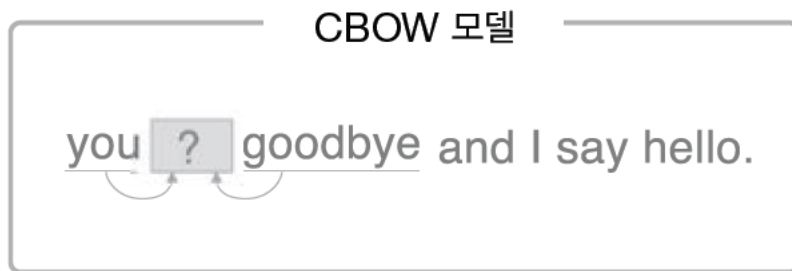
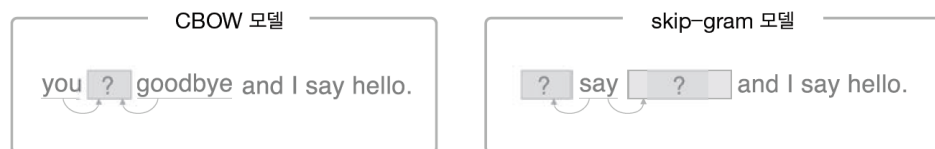


그림 3-23 CBOW 모델과 skip-gram 모델이 다루는 문제



-CBOW vs skip-gram

그림 3-23 CBOW 모델과 skip-gram 모델이 다루는 문제



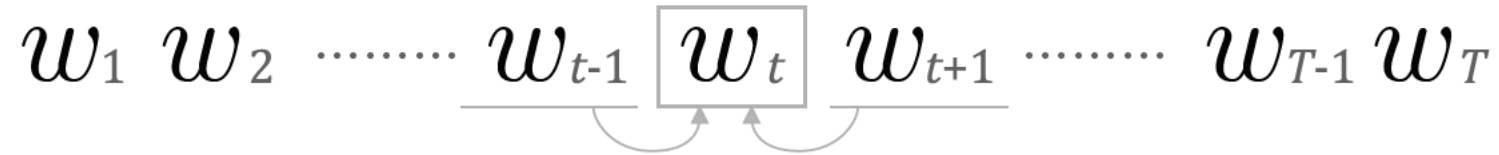
- 한 번의 학습으로 단어 하나에 대한 학습을 하는 CBOW
- 한 번의 학습으로 양쪽 문맥 모두 학습하는 Skip-gram
- Skip-gram은 은닉층 평균을 구하지 않는다.

Source Text	Training Samples
The quick brown fox jumps over the lazy dog. →	(the, quick) (the, brown)
The quick brown fox jumps over the lazy dog. →	(quick, the) (quick, brown) (quick, fox)
The quick brown fox jumps over the lazy dog. →	(brown, the) (brown, quick) (brown, fox) (brown, jumps)
The quick brown fox jumps over the lazy dog. →	(fox, quick) (fox, brown) (fox, jumps) (fox, over)



-모델과 확률

그림 3-22 word2vec의 CBOW 모델(맥락의 단어로부터 타깃 단어를 추측)



$P(w_t | w_{t-1}, w_{t+1})$ w_{t-1}, w_{t+1} 이 주어졌을 때
 w_t 가 일어날 확률

$$L = -\frac{1}{T} \sum_{t=1}^T \log P(w_t | w_{t-1}, w_{t+1})$$

CBOW

$$L = -\frac{1}{T} \sum_{t=1}^T (\log P(w_{t-1} | w_t) + \log P(w_{t+1} | w_t))$$

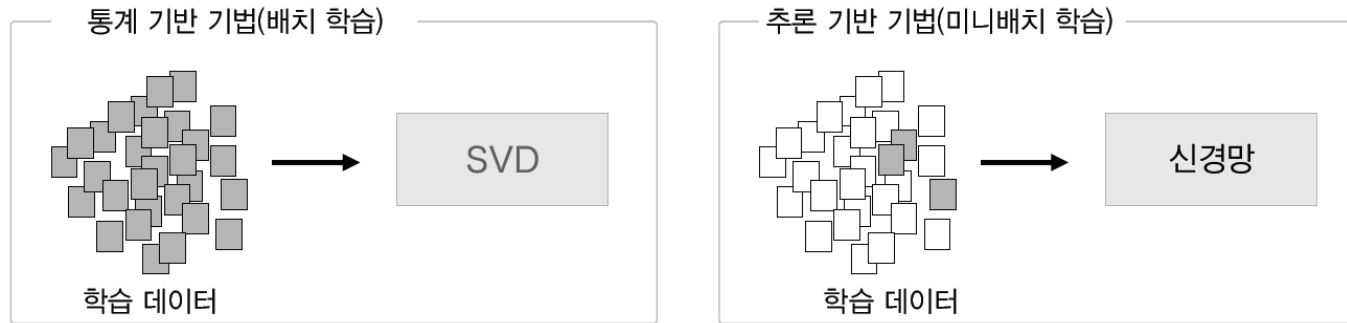
Skip-gram

L을 최소화 시키는 것이 모델의 목표



-통계 기반 vs 추론 기반

그림 3-1 통계 기반 기법과 추론 기반 기법 비교



- 추론 기반 기법은 병렬 처리로 학습 속도 높일 수 있다
- 통계 기반 기법은 단어 분산표현 갱신 필요 시 전체를 다 다시 학습시켜야 하는 한계가 있음



Thanks!

