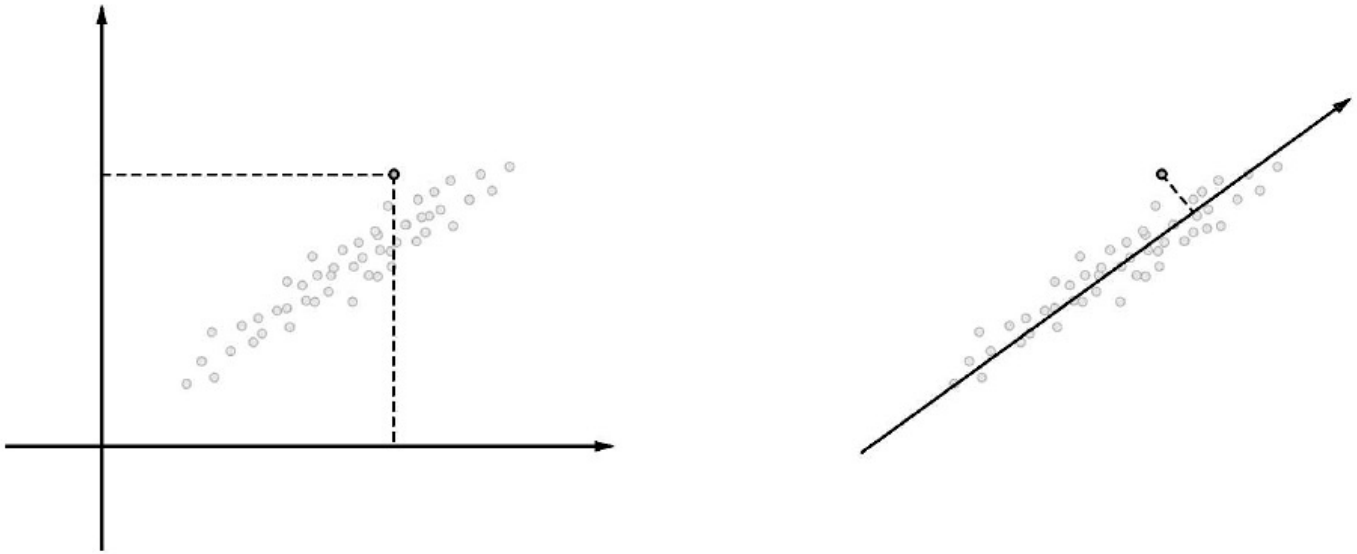
Word2vec

2가지 Architecture 모델 수학적으로 이해하기

review (+ 차원감소)

통계기반기법

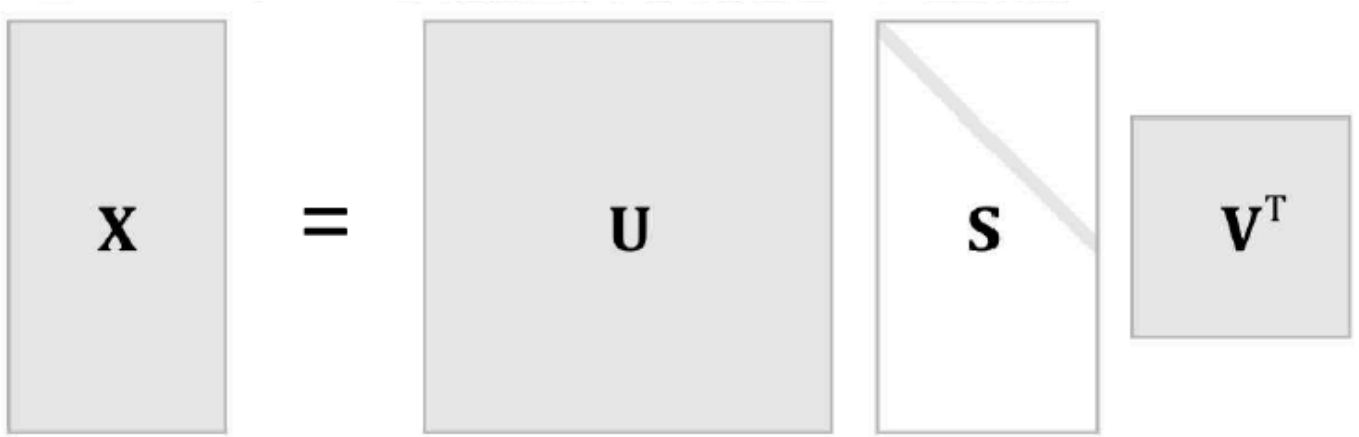
- 분포가설 / 단어의 분산 표현
- 윈도우 크기
- 동시발생 행렬 / PPMI 행렬
- **차원 감소**



특이값 분해

Singular Value Decomposition(SVD)

$$X = USV^T$$



U, V : 직교행렬 (orthogonal matrix)
S : 대각행렬 (diagonal matrix)

특이값 분해

Singular Value Decomposition(SVD)

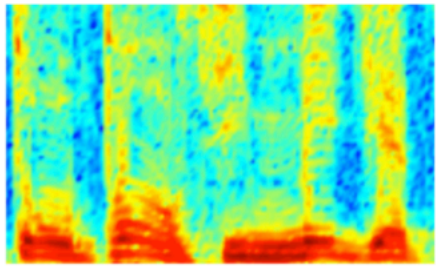
$$X = USV^T$$

U, V : 직교행렬 (orthogonal matrix)
S : 대각행렬 (diagonal matrix)

Word Embeddings

: vector representations of words

AUDIO



Audio Spectrogram

DENSE

IMAGES

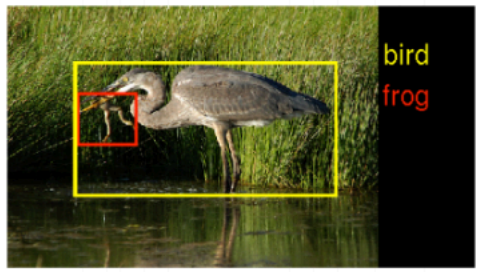
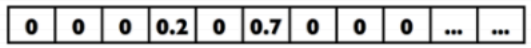


Image pixels

DENSE

TEXT



Word, context, or
document vectors

SPARSE

Problems with this discrete representation

The vast majority of rule-based and statistical NLP work regards words as atomic symbols: hotel, conference, walk

In vector space terms, this is a vector with one 1 and a lot of zeroes

[0 0 0 0 0 0 0 0 0 0 1 0 0 0 0]

Dimensionality: 20K (speech) – 50K (PTB) – 500K (big vocab) – 13M (Google 1T)

Word meaning is defined in terms of vectors

We will build a dense vector for each word type, chosen so that it is good at predicting other words appearing in its context

… those other words also being represented by vectors … it all gets a bit recursive

$$\textit{linguistics} = \begin{pmatrix} 0.286 \\ 0.792 \\ -0.177 \\ -0.107 \\ 0.109 \\ -0.542 \\ 0.349 \\ 0.271 \end{pmatrix}$$

출처: Stanford University NLP lecture2

추론 기반 기법

Word2vec

단어 임베딩(word embedding)을 생성하는 관련된 모델들

Tomas Mikolov
Google Inc.

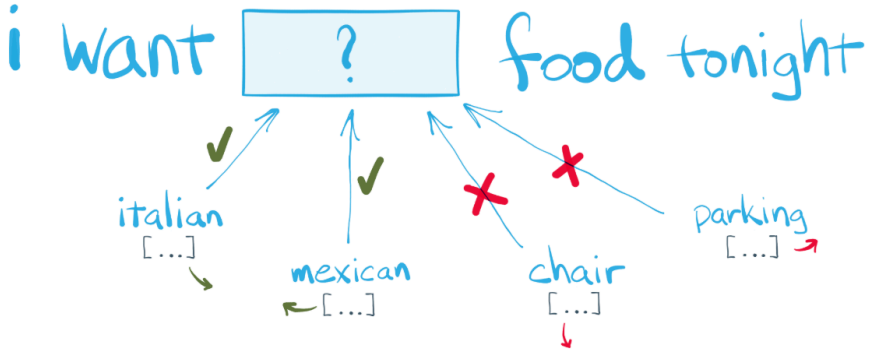
단순한 2층 layer 신경망 모델

- **CBOW** (continuous bag of words)
- **Skip-gram**

Q Word2vec은 단어를 표현하는 방법을 어떻게 학습하는 것일까?

단어의 주변을 보면 그 단어를 안다. You shall know a word by the company it keeps.
– 언어학자 J.R. Firth (1957)

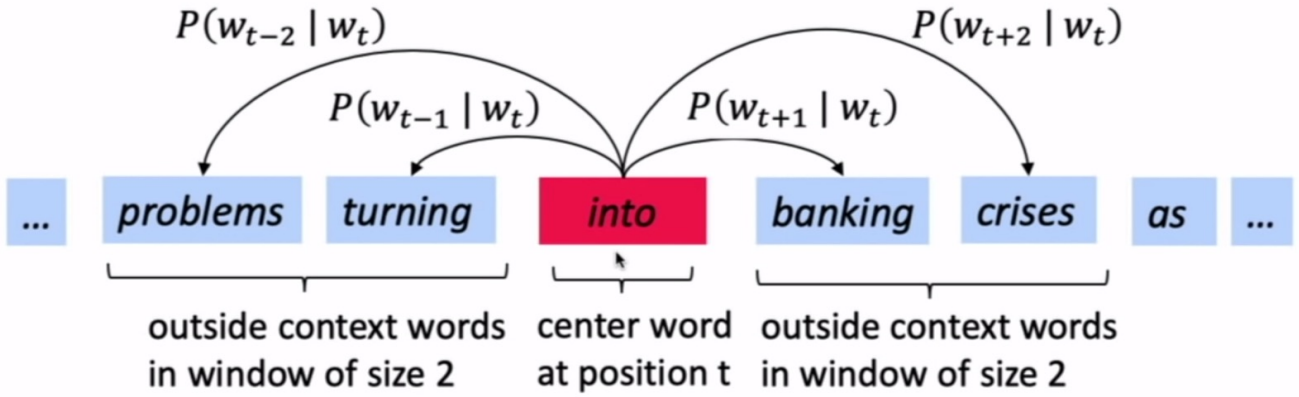
빈칸에 어떤 단어가 들어갈 수 있을까?



빈칸에 들어가기 적합한 단어들과 부적합한 단어들

Word2Vec Overview

- Example windows and process for computing $P(w_{t+j} \mid w_t)$



출처: Stanford University NLP lecture2

확률 관점

CBOW 모델이 하는 일 : 맥락을 주면 타깃 단어가 출현할 확률을 출력하는 것

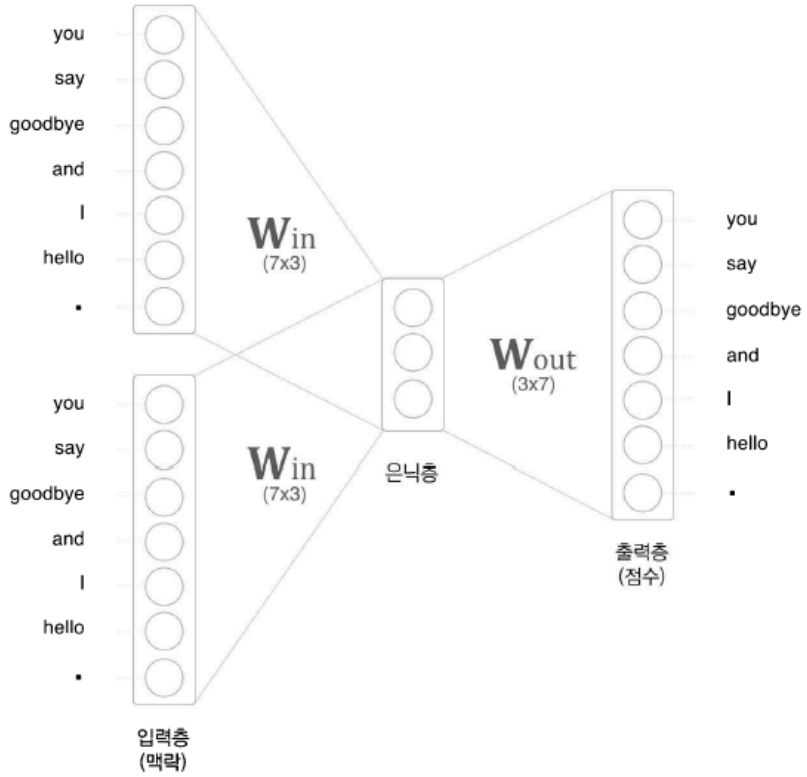
$$w_1 \; w_2 \; \cdots\cdots \; w_{t-1} \; \boxed{w_t} \; w_{t+1} \; \cdots\cdots \; w_{T-1} \; w_T$$

- 맥락 W t-1 과 W t+1 로주어졌을 때 타깃이 Wt가 될 확률

★ $P(w_t \mid w_{t-1}, w_{t+1})$

- 음의 로그 가능도 (negative log likelihood)

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



이때의 가중치 매개변수가 우리가 얻고자 하는 단어의 분산 표현!

단어의 분산 표현을 생성하는 두가지 아키텍쳐 모델
CBOW 와 skip grams

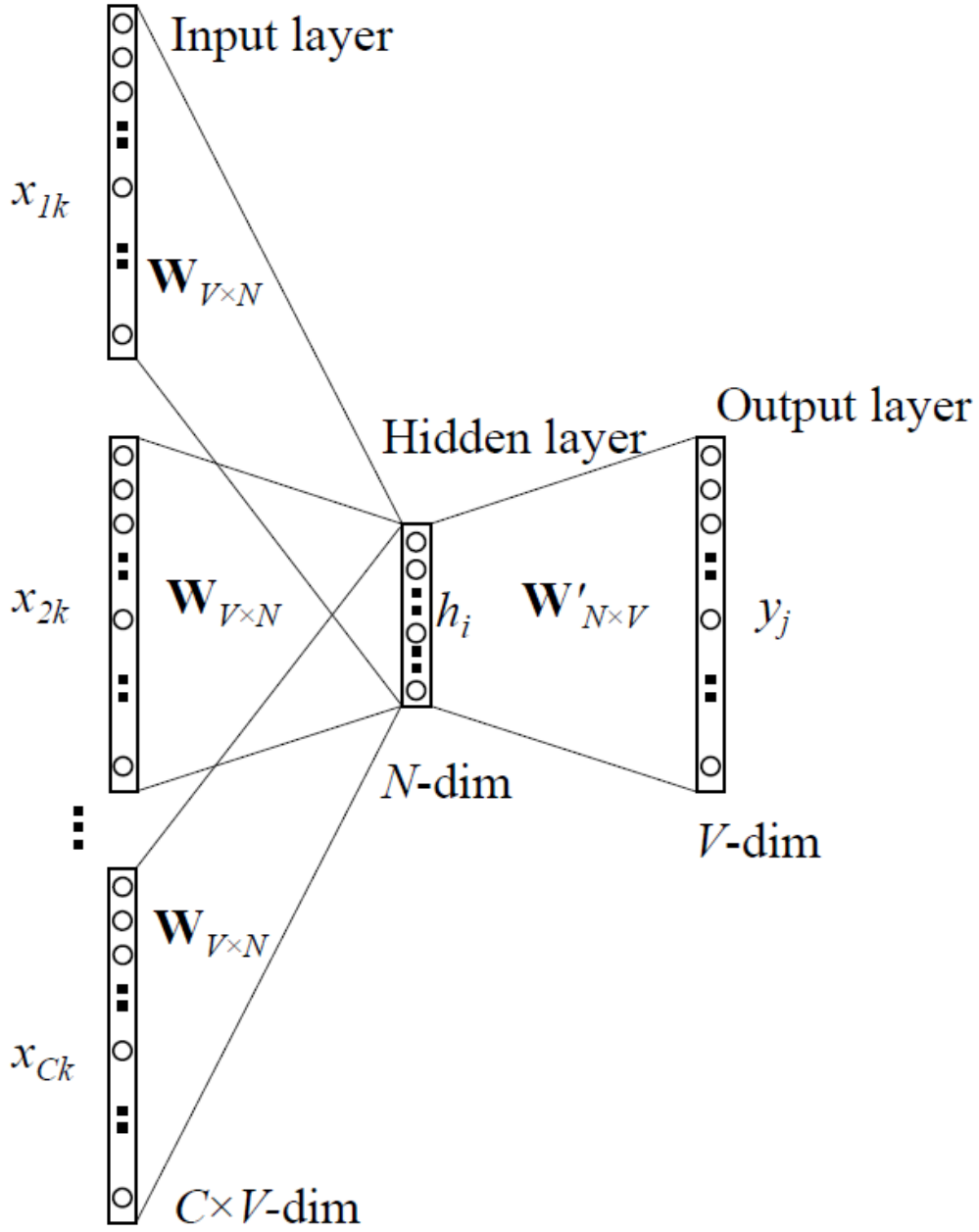
1. continuous bag-of-words architecture

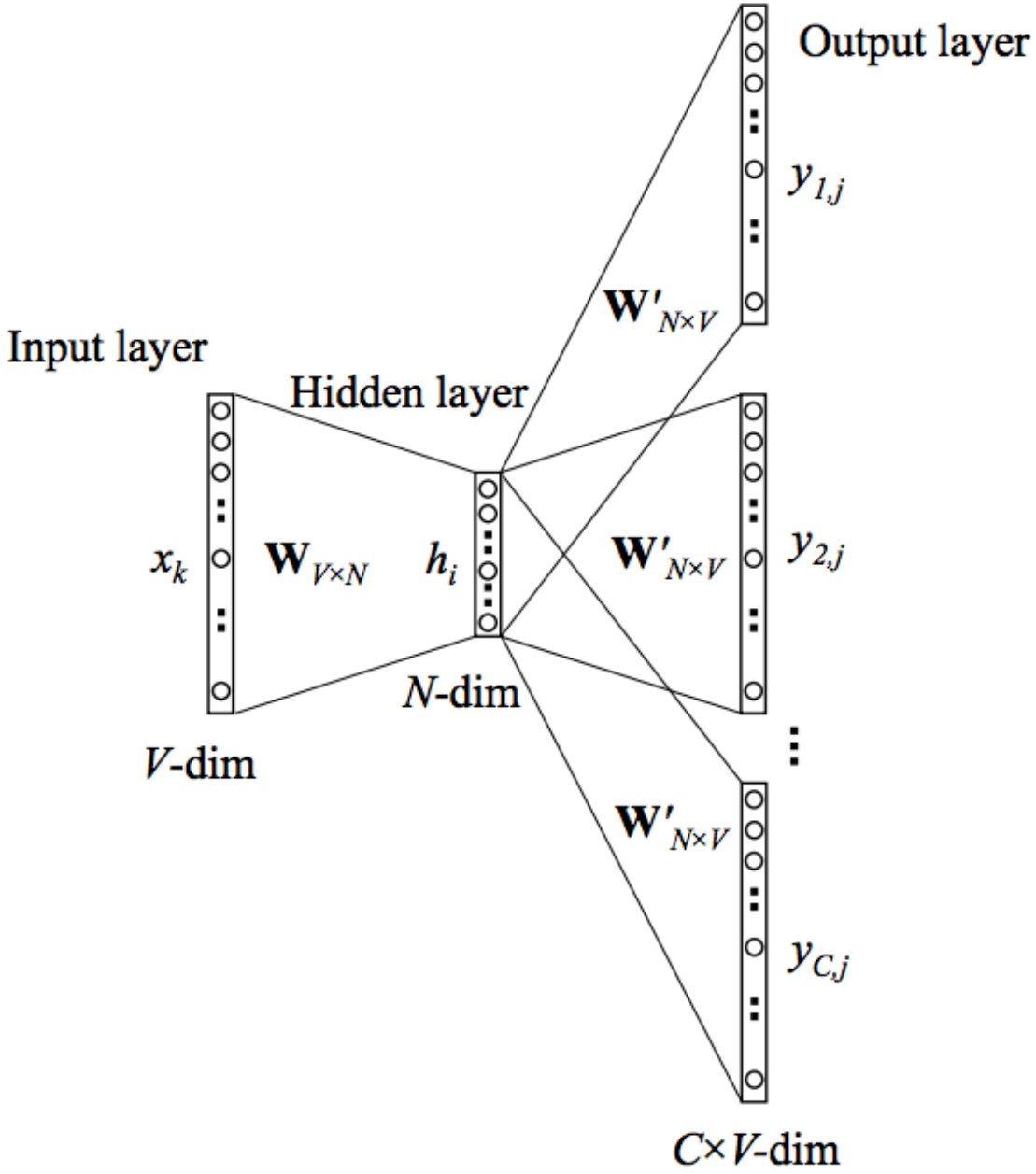
: predicts the current word from a window of surrounding context words

2.skip grams architecture

: uses the current word to predict the surrounding window of context words

CBOW Architecture

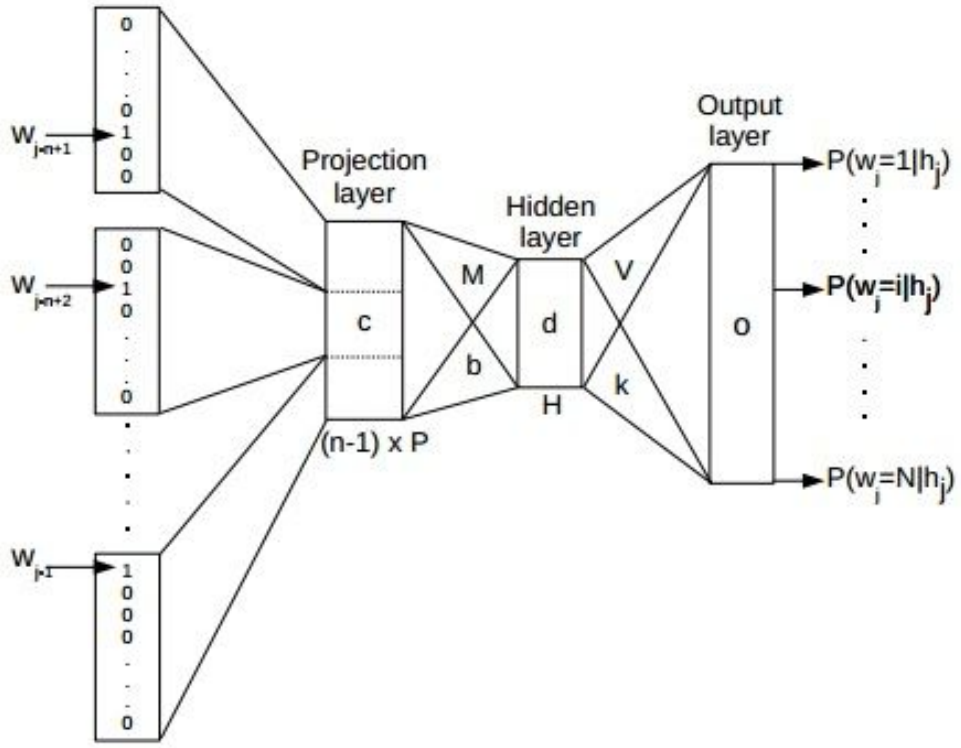




Skip-gram Architecture

모델의 계산복잡도 비교

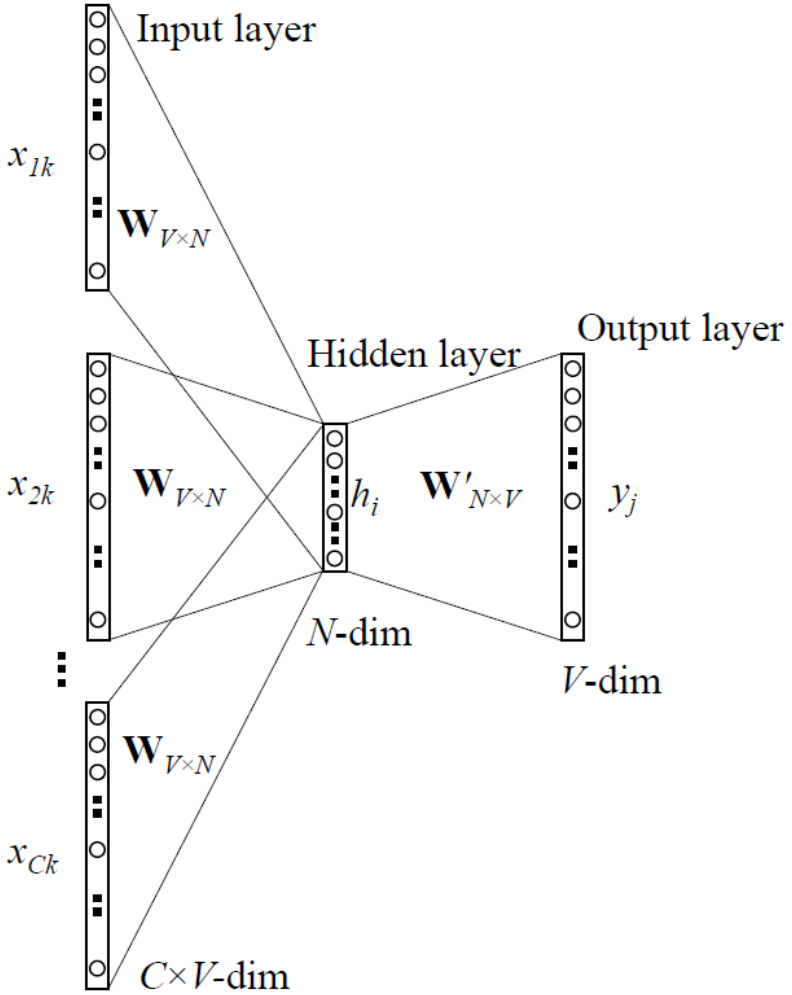
Feed-Forward Neural Net Language Model (NNLM)



- 입력층 (단어들 Projection) - N x P
- 은닉층 (Projection Layer에서 Hidden Layer로) - N x P x H
- 출력층 (Hidden Layer에서 Output Layer로) - H x V

NxP + NxPxH + HxV

ex) N=10, P=500, H=500 일때 O(HxV) = O(**50억**)



CBOW 모델은?

- 입력층(C개 단어 Projection) - C x N
- 은닉층 (Projection Layer에서 Output Layer) - N x V

CxN + NxV
(V를 ln V로 -> **CxN + N x lnV**)

ex) C=10, N=500, V=1,000,000으로 잡아도
O(500 x (10+ln(1,000,000))) = 약 (**10000**)

Next

RNN

Reference

밑바닥부터 시작하는 딥러닝2
Stanford University Lecture 2 | Word Vector Representations: word2vec
Tensorflow.org | Vector representations of words
beomsu kim의 블로그 'word2vec 관련 이론정리'
Dreamgonfly의 블로그 '쉽게 씌여진 word2vec'
于 航의 블로그 https://zhuanlan.zhihu.com/p/43736169