# Word Embedding

- Word2vec

Most of this material is from JH Lee at AI Lab. HYU

## Natural Language Processing

| | |
|---|---|
| Text Classification | 호날두가 골을 넣었습니다 → 스포츠 |
| Language Modeling | 나는 학교에 ( ? ) → 간다, 왔다, … |
| Question Answering | 링컨은 언제 태어났습니까? → 1809년 |
| Machine Translation | 좋은 아침입니다 → Good Morning |
| Named Entity Recognition | 문재인은 대한민국의 대통령이다 → PER(문재인), LOC(대한민국) |

공통점 : 모델이 다루어야 할 데이터가 모두 **Text** 이다.

# Image & Text to Numeric Data

⇨

| | | | |
|---|---|---|---|
| 0.3 | 0.2 | 0.6 | 0.1 |
| 0.2 | 0.8 | 0.3 | 0.5 |
| 0.1 | 0.9 | 0.4 | 0.4 |
| 0.1 | 0.7 | 0.6 | 0.3 |

⇨ Model ⇨ Dog

호날두가 골을 넣었습니다 ⇨ ??? ⇨ Model ⇨ 스포츠

# Word Embedding

**정의**

- 단어(words)를 실수의 벡터(vectors of real numbers)로 매핑시키는 것
- 매핑된 벡터를 **Word Representation**이라고도 함

**방법론**

- One-hot Encoding
- Latent Sentiment Analysis
- NNLM, RNNLM (Bengio, 2003)
- Word2vec (Mikolov; Google, 2013)
- GloVe (Pennington; Stanford University, 2014)
- FastText (Bojanowski; Facebook, 2016)

# One-hot Encoding

호날두가 골을 넣었습니다

| | | | |
|---|---|---|---|
| 호날두 | 1 | 0 | 0 |
| 골 | 0 | 1 | 0 |
| 넣다 | 0 | 0 | 1 |

**Numerical Data**

⇨ Model

# One-hot Encoding

호날두가 골을 넣었습니다
메시도 골을 넣었습니다
류현진이 호투했습니다
…

**Sparse Vectors** $\in \mathbb{R}^{|V|}$

| | | | | | | |
|---|---|---|---|---|---|---|
| 호날두 | 1 | 0 | 0 | 0 | ... | 0 |
| 골 | 0 | 1 | 0 | 0 | ... | 0 |
| 넣다 | 0 | 0 | 1 | 0 | ... | 0 |
| 메시 | 0 | 0 | 0 | 1 | ... | 0 |
| … | | | … | | | |
| 호투 | 0 | 0 | 0 | 0 | ... | 1 |

**Curse of Dimensionality**

# One-hot Encoding

호날두가 골을 넣었습니다

| | | | |
|---|---|---|---|
| 호날두 | 1 | 0 | 0 |
| 골 | 0 | 1 | 0 |
| 넣다 | 0 | 0 | 1 |

호날두 · 골 = 0

$dist(\text{호날두}, \text{골}) = dist(\text{골}, \text{넣다})$

모든 벡터가 Orthogonal 하고 거리가 같음 → 서로 Independent

→ **연관성, 유사성을 표현할 수 없음**

# Distributed Representation

**Problems of One-hot Encoding**

- Sparsity
- Independence

**Solution**

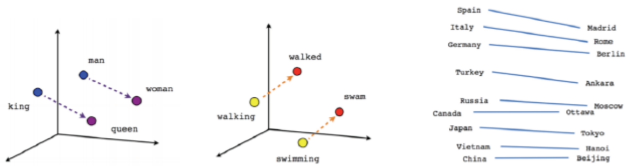
- Distributed Representations
- 연속적이고 작고 빽빽한(dense) 벡터로 표현
- $\mathbb{R}^{|V|} \rightarrow \mathbb{R}^{|n|}$
  $|V| \gg n$

| | | |
|---|---|---|
| 호날두 | 0.7 | 0.2 |
| 골 | 0.6 | 0.7 |
| 넣다 | 0.2 | 0.8 |

# Distributed Representation

## Word2vec

- Distributed Representation을 만드는 Word Embedding Model
- 2013년 Google(Mikolov et al.)에서 제안
- 단어 간의 의미적 관계를 고려한 Representation을 생성
- 이전 모델들(LSA, NNLM 등)보다 계산복잡도를 줄이고, 성능을 높임

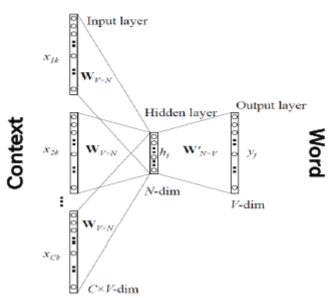



Male-Female    Verb tense    Country-Capital

# Word2vec

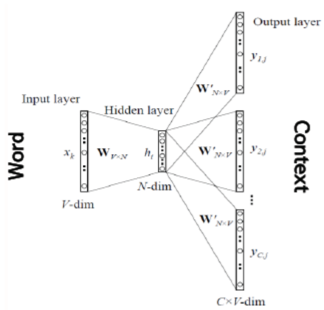
## Continuous Bag-of-Words (CBOW)

- 나는 ___에 간다



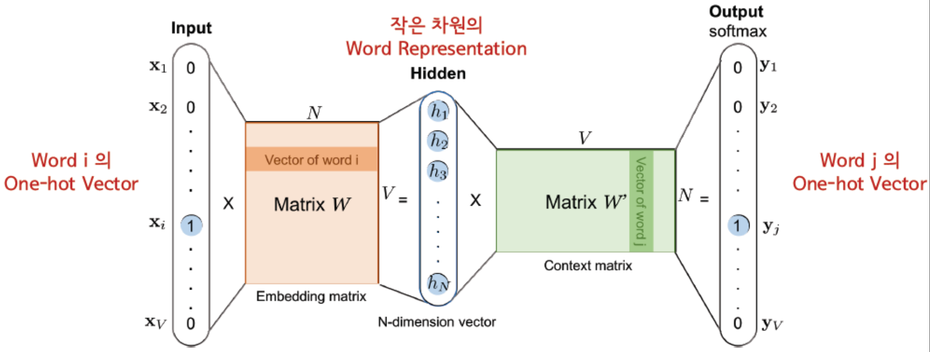

## Continuous Skip-gram (Skip-gram)

- ___는 외나무다리에서 ___



# Word2vec

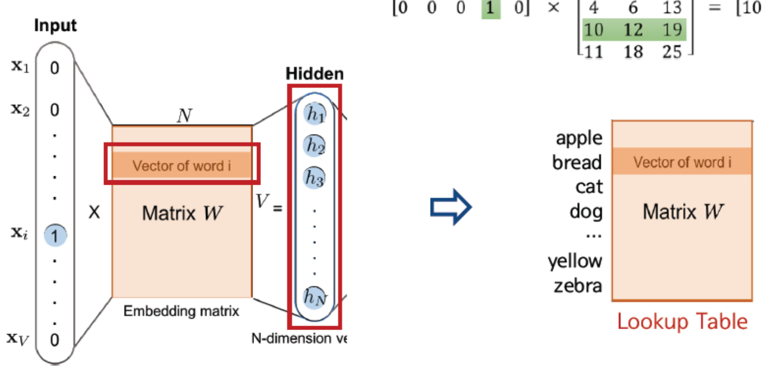
- Continuous Bag-of-Words (CBOW)
- Continuous Skip-gram (Skip-gram)




# Word2vec

- Lookup Table

$$[0 \quad 0 \quad 0 \quad 1 \quad 0] \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} = [10 \quad 12 \quad 19]$$



## Skip-gram

- the quick brown fox jumped over the lazy dog

  * Window size = 1

  → (quick, the), (quick, brown)

## Skip-gram

- the quick brown fox jumped over the lazy dog

  * Window size = 1

  → (quick, the), (quick, brown)
  → (brown, quick), (brown, fox)

# Skip-gram

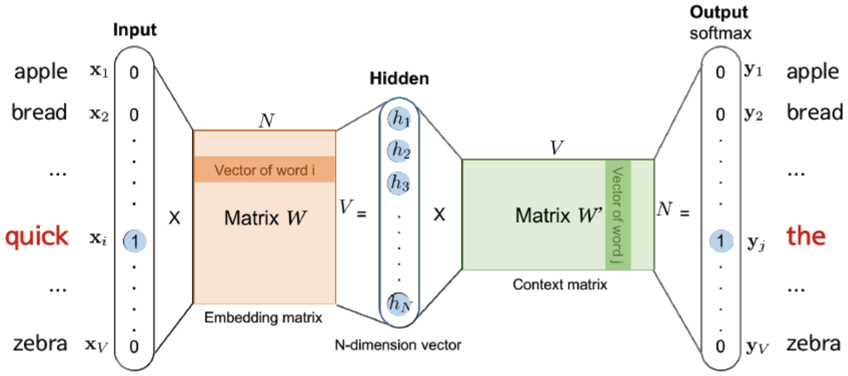
- the quick brown fox jumped over the lazy dog

* Window size = 1

→ (quick, the), (quick, brown)
→ (brown, quick), (brown, fox)
→ (fox, brown), (fox, jumped)
→ …

# Skip-gram

- the quick brown fox jumped over the lazy dog



# Skip-gram

**Objective Function**

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-s\le j\le s, j\neq 0}\log p(w_{t+j}|w_t) \qquad (\ s = \text{window size}\ ) \qquad (1)$$

$$P(o|c) = \frac{exp(u_o^T v_c)}{\sum_{w=1}^{V} exp(u_w^T v_c)} \qquad (2)$$

- Equation 1을 최대화하는 방향으로 학습
- Equation 1의 조건부확률은 Equation 2의 Softmax 함수를 통해 구할 수 있음
- $c$ , $o$ 는 Input, Output word를 의미
- $v_c$ , $u_o$ 는 $w$에 대한 Input, Output의 Representation Vector를 의미

a word

# Skip-gram

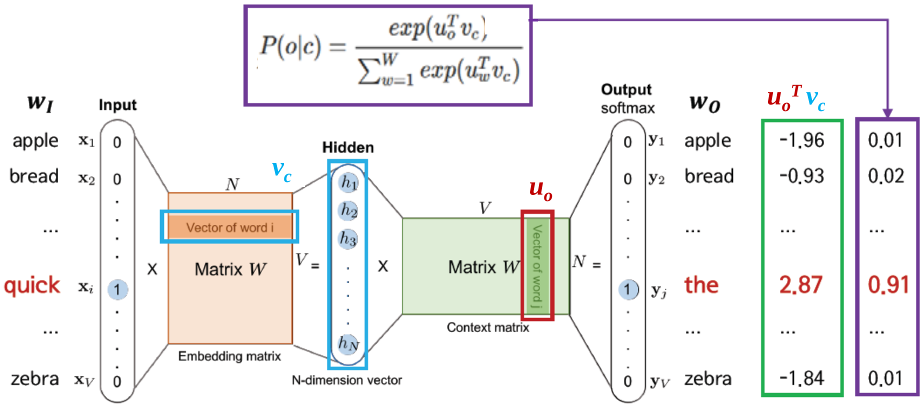
$$P(o|c) = \frac{exp(u_o^T v_c)}{\sum_{w=1}^{V} exp(u_w^T v_c)}$$

$$\begin{aligned}
\frac{\partial}{\partial v_c}\ln P(o|c) &= \frac{\partial}{\partial v_c}\ln\frac{exp(u_o^T v_c)}{\sum_{w=1}^{V} exp(u_w^T v_c)} \\
&= \frac{\partial}{\partial v_c}u_o^T v_c - \frac{\partial}{\partial v_c}\ln\sum_{w=1}^{V} exp(u_w^T v_c) \\
&= u_o^T - \frac{1}{\sum_{w=1}^{V} exp(u_w^T v_c)}\left(\sum_{w=1}^{V} exp(u_w^T v_c)\cdot u_w^T\right) \\
&= u_o^T - \sum_{w=1}^{V}\frac{exp(u_w^T v_c)}{\sum_{w=1}^{W} exp(u_w^T v_c)}\cdot u_w^T \\
&= u_o^T - \sum_{w=1}^{V} P(w|c)\cdot u_w^T
\end{aligned}$$

update rule for $v_c$ → $v_c^{t+1} = v_c^t + \alpha(u_o - \sum_{w=1}^{V} P(w|c)\cdot u_w)$

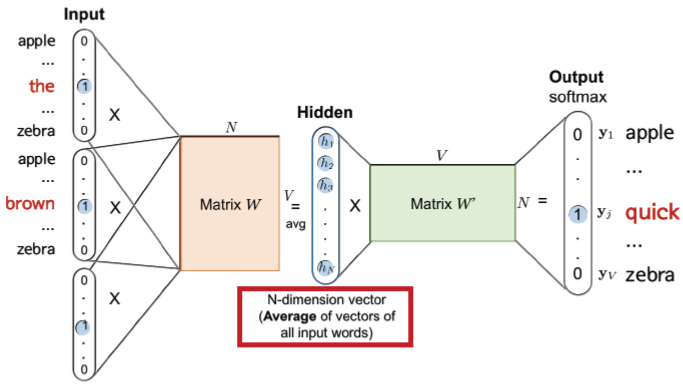
update rule for $u_o$ → just swap $v_c$ ***and*** $u_o$ (= derivative of $ln\ P(o|c)$ in terms of $u_o$)

# Skip-gram




# CBOW

- the quick brown fox jumped over the lazy dog



# Conclusion and Future Works

**Conclusion**

- Word Embedding이란, 단어를 Vector로 매핑하는 것
- Word2vec은 Embedding Model 중 하나
- 함께 나타난 (co-occurrence) 주변 단어와의 의미적 관계를 학습
- 대부분의 NLP Model에서 pre-trained embedding으로 사용

**Future Works**

- Negative Sampling
- Other Embedding Models (GloVe, FastText)
- Contextual Embedding Models (ELMo, ULMFiT)