Word2vec

壽險程設科

崔嘉祐

Outline

- NLP (Natural Language Processing)
- Vector Space of Semantics
- Word2vec

NLP

- 是 AI 和 Linguistics 學科 。從1950年代開始，此領域探討如何處理及運用自然語言；自然語言認知則是指讓電腦懂人類語言。

- 1980年開始語言處理開始使用機器學習的演算法。

NLP

- 語音識別（Speech recognition）
- 信息檢索（Information retrieval）
- 問答系統（Question answering）
- 機器翻譯（Machine translation）
- 自動摘要（Automatic summarization）

NLP

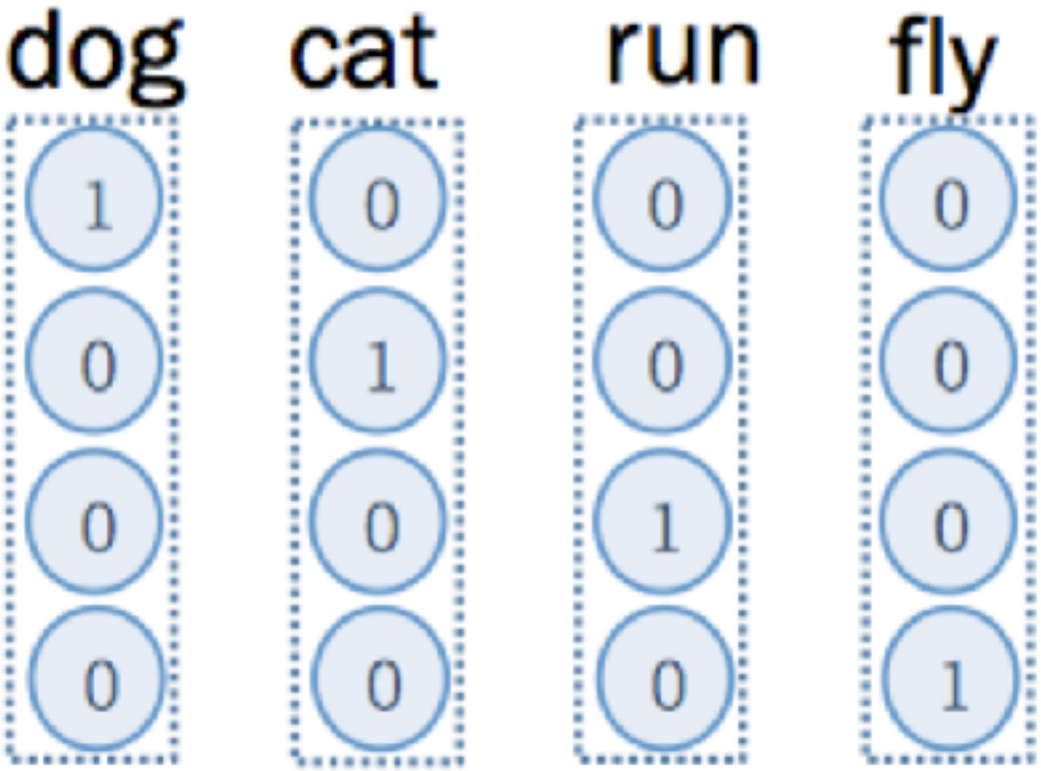
- Problem
  - 單詞的邊界界定 (e.g. 全台大停電)
  - 詞義的消歧 (e.g. 水)
  - 句法的模糊性
  - 有瑕疵的或不規範的輸入
  - 語言行為與計劃

Outline

- NLP (Natural Language Processing)
- Vector Space of Semantics
  - One-hot-Encoding
  - Context-based
- Word2vec

Vector Space of Semantics

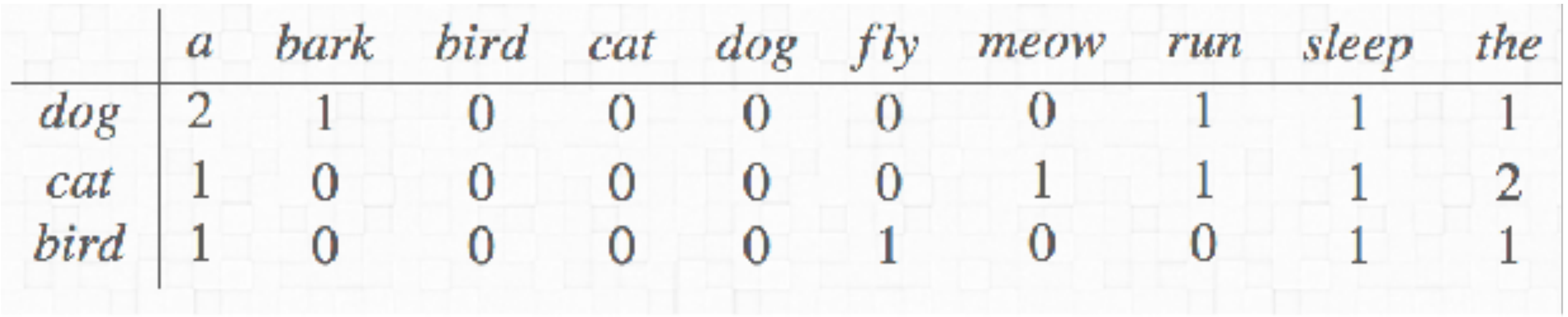
- One-hot-Encoding : 假設每個字的語意是不相干的。也就是說，每個字的向量都是互相垂直。



Vector Space of Semantics

- Context-based : 以上下文的向量表示。

1. The dog run.
2. A cat run.
3. A dog sleep.
4. The cat sleep.
5. A dog bark.
6. The cat meows.
7. The bird fly.
8. A bird sleep.



	a	*bark*	*bird*	*cat*	*dog*	*fly*	*meow*	*run*	*sleep*	*the*
dog	2	1	0	0	0	0	0	1	1	1
cat	1	0	0	0	0	0	1	1	1	2
bird	1	0	0	0	0	1	0	0	1	1

Vector Space of Semantics

- Euclidean distance

$$\begin{aligned}\mathrm{d}(\mathbf{p},\mathbf{q}) = \mathrm{d}(\mathbf{q},\mathbf{p}) &= \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} \\ &= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.\end{aligned}$$

- Cosine similarity

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2} \sqrt{\sum\limits_{i=1}^{n} B_i^2}}$$

Outline

- NLP (Natural Language Processing)
- Vector Space of Semantics
- Word2vec
    - Skip-gram
    - CBOW

Word2vec

- Distributional hypothesis (分布假說)
- Word Embedding (representation, vector)
- Solve the curse of dimensionality (詞彙量大)

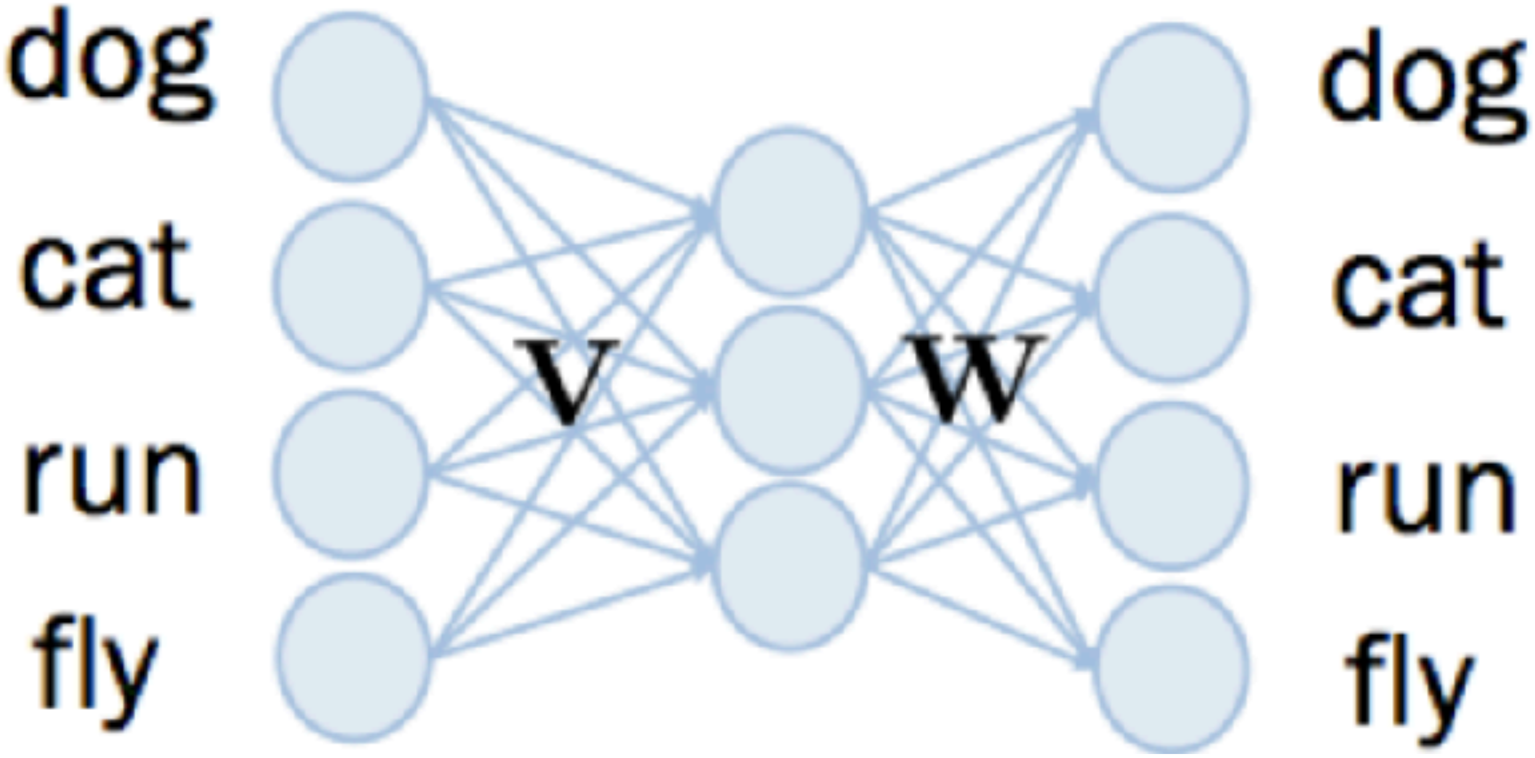
Word2vec

- Network architecture

Input layer

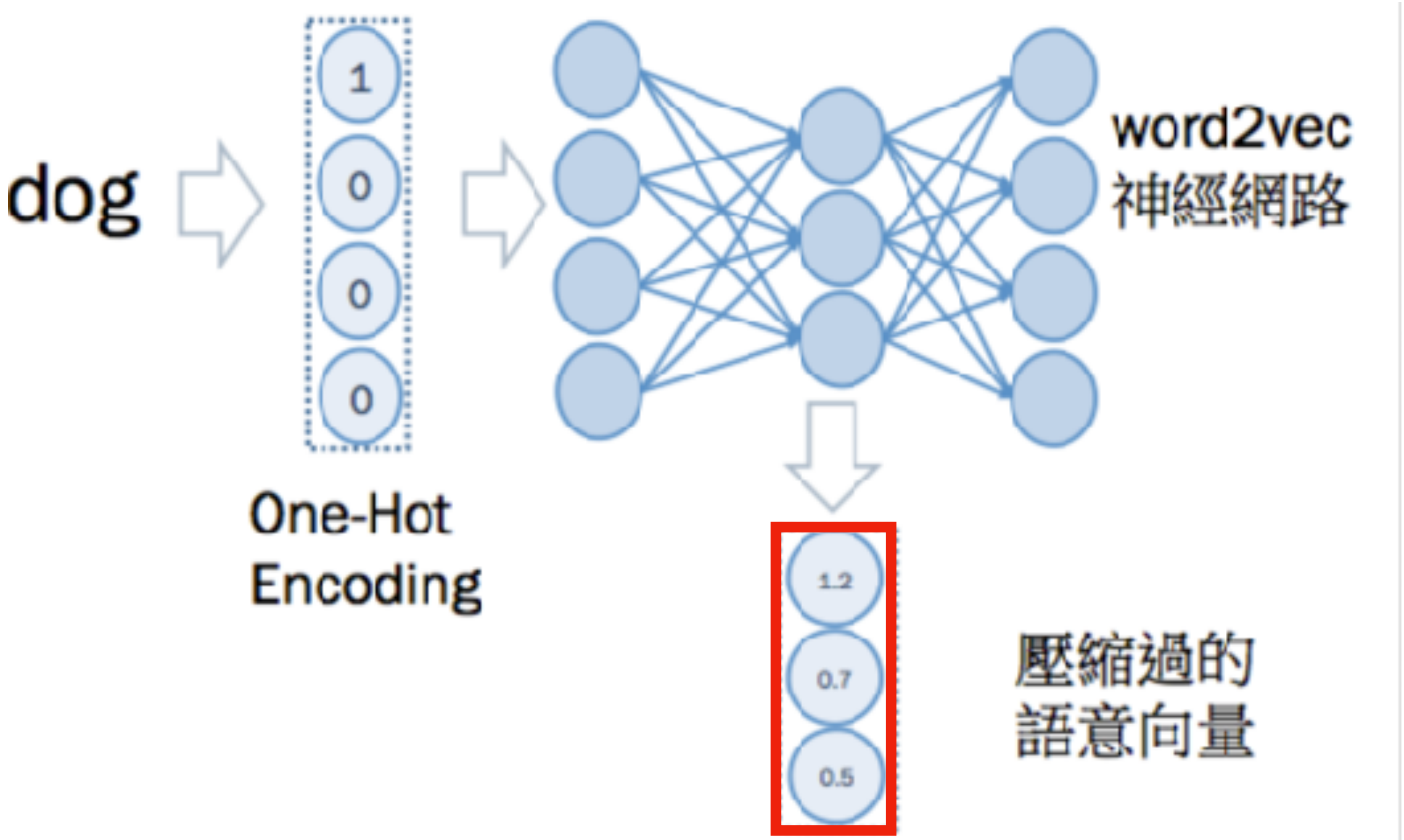
Hidden layer

Output layer



Word2vec

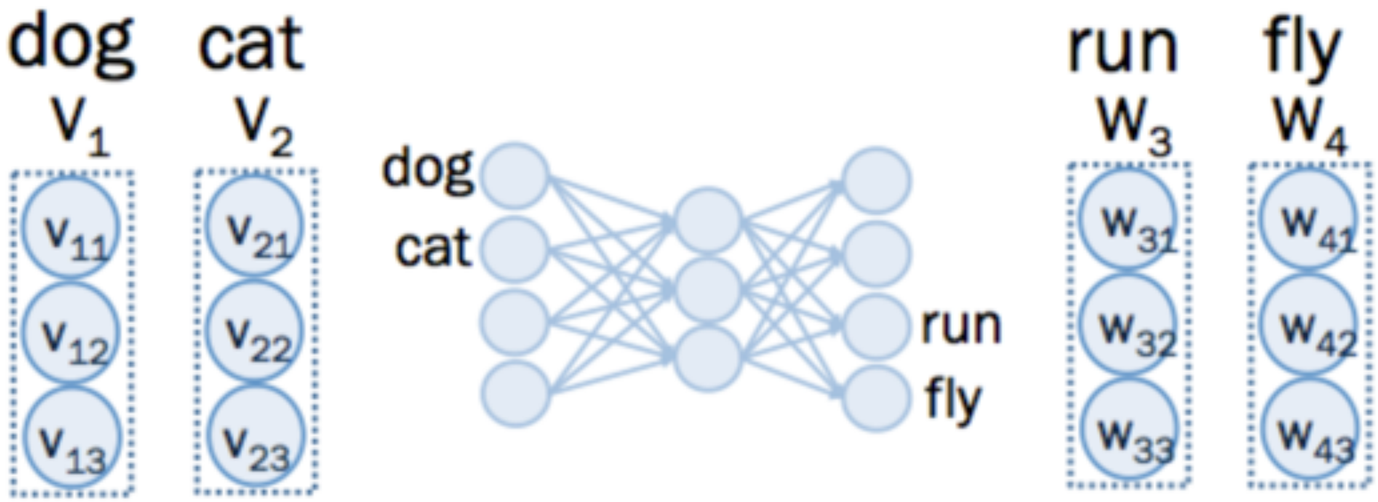
- 矩陣相乘

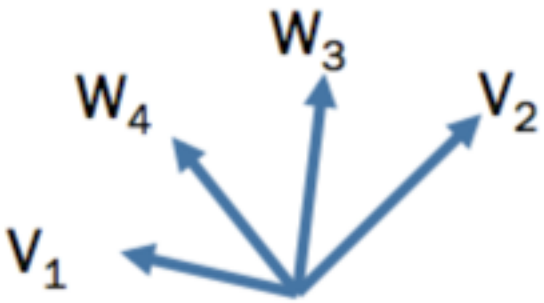


$$\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_{11} & v_{12} & v_{13} \\ v_{21} & v_{22} & v_{23} \\ v_{31} & v_{32} & v_{33} \\ v_{41} & v_{42} & v_{43} \end{bmatrix} = \begin{bmatrix} v_{11} & v_{12} & v_{13} \end{bmatrix}$$

Word2vec

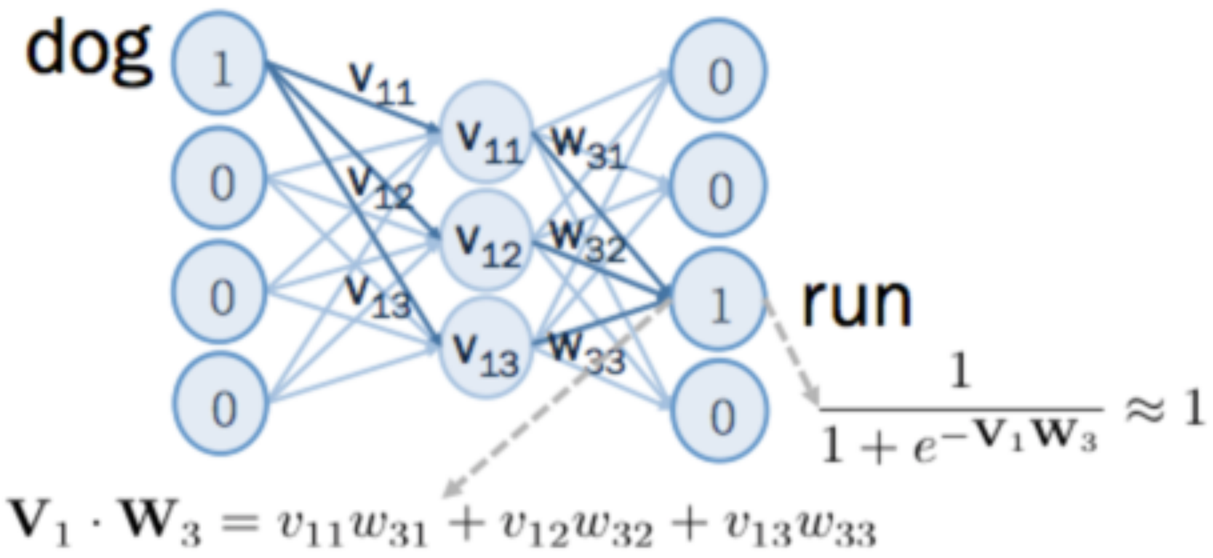
- 向量初始化（V1, V2, W3, W4）





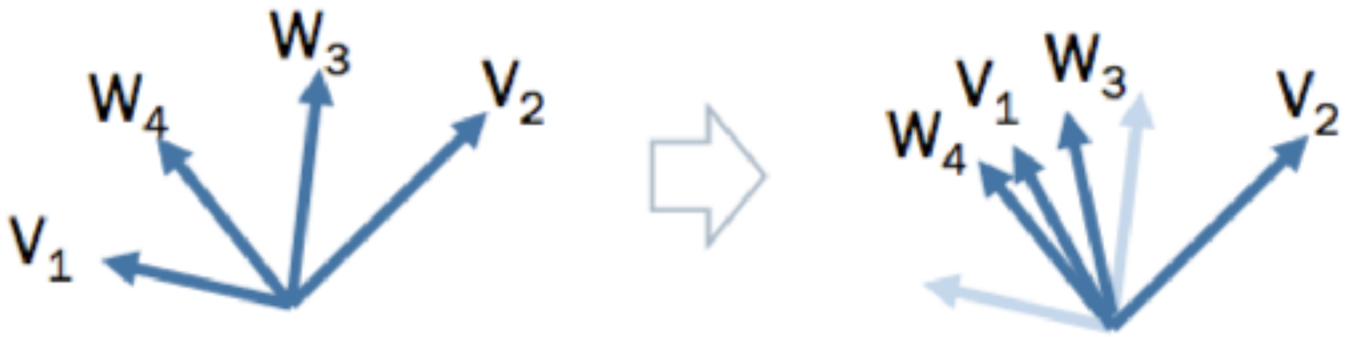
Word2vec

- Backward propagation



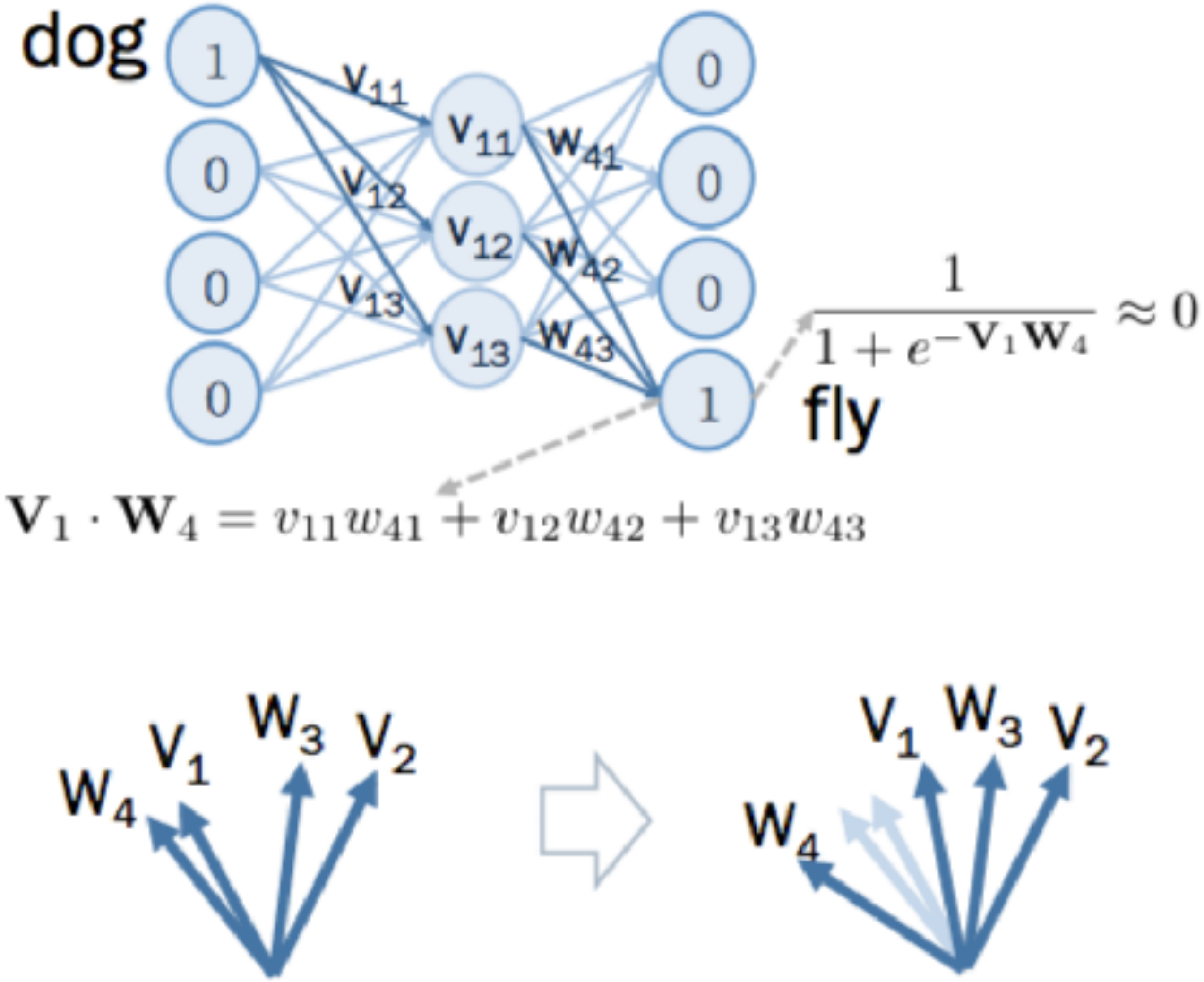
$$\frac{1}{1+e^{-\mathbf{V}_1\mathbf{W}_3}} \approx 1$$

$$\mathbf{V}_1 \cdot \mathbf{W}_3 = v_{11}w_{31} + v_{12}w_{32} + v_{13}w_{33}$$



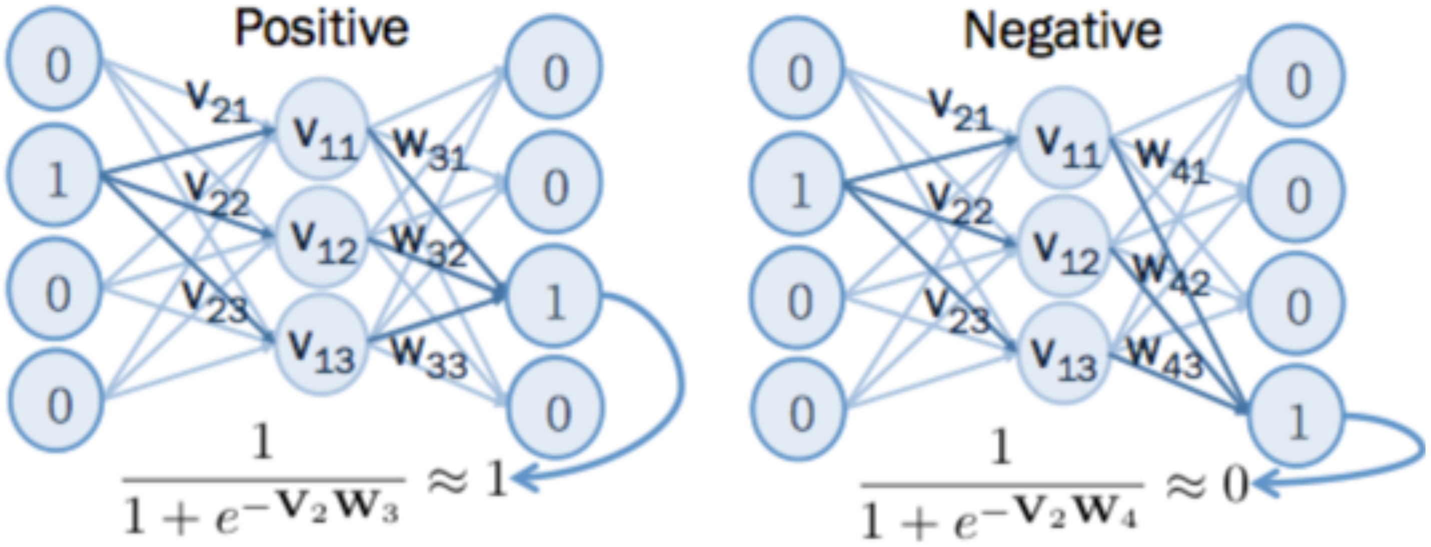
Word2vec

- Backward propagation


dog
1
0
0
0
v_{11}
v_{12}
v_{13}
v_{11}
v_{12}
v_{13}
w_{41}
w_{42}
w_{43}
0
0
0
1
fly
$\frac{1}{1+e^{-\mathbf{V}_1\mathbf{W}_4}} \approx 0$
$\mathbf{V}_1 \cdot \mathbf{W}_4 = v_{11}w_{41} + v_{12}w_{42} + v_{13}w_{43}$
W_4
V_1
W_3
V_2
V_1
W_3
V_2
W_4

Word2vec

- Backward propagation



Word2vec

- Objective function

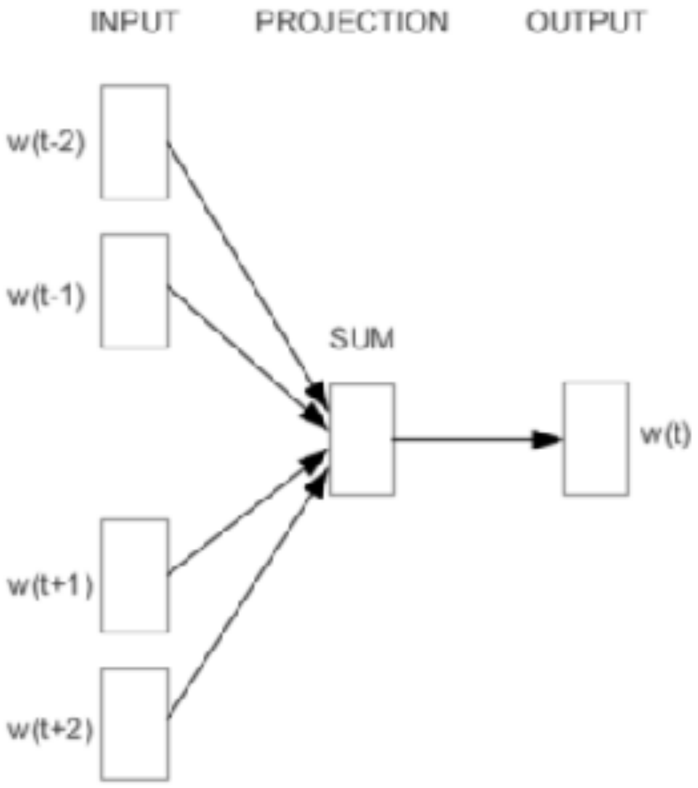
$$J = -\log(\boxed{\frac{1}{1+e^{-v_I \cdot w_{pos}}}}) - \sum_{neg} \log(1 - \boxed{\frac{1}{1+e^{-v_I \cdot w_{neg}}}})$$

目標趨近於 1　　**目標趨近於 0**

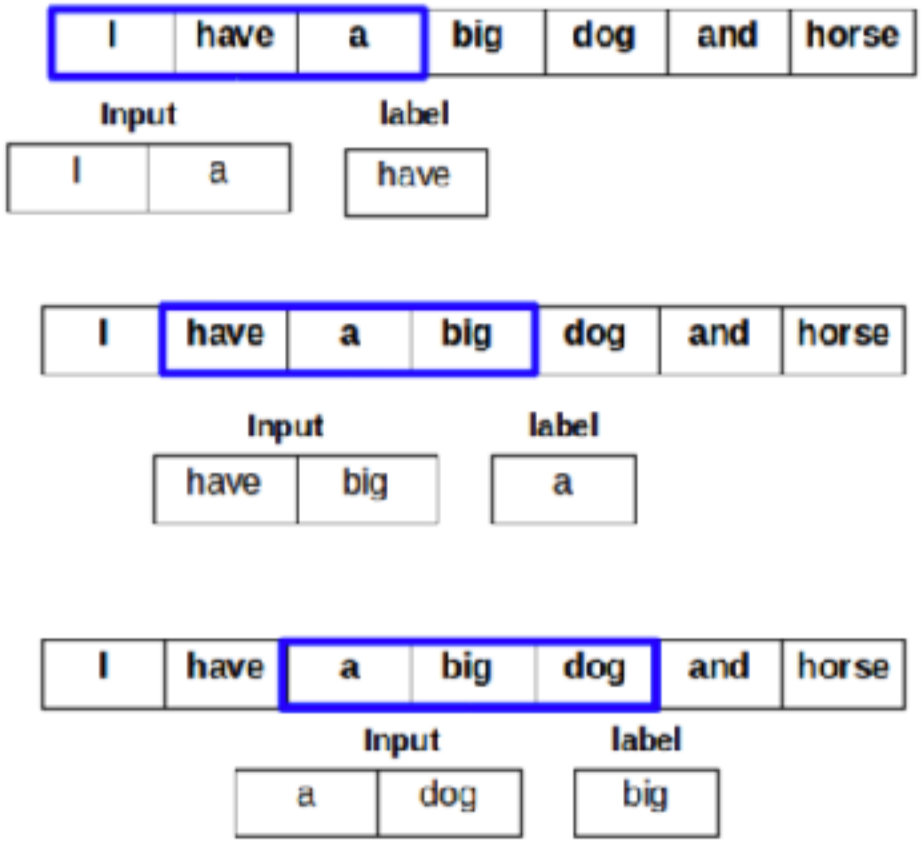
- 目標 minimize function “J”

Word2vec

- CBOW (Continuous Bag of Word)



CBOW



I	have	a	big	dog	and	horse

Input: I, a
label: have

I	have	a	big	dog	and	horse

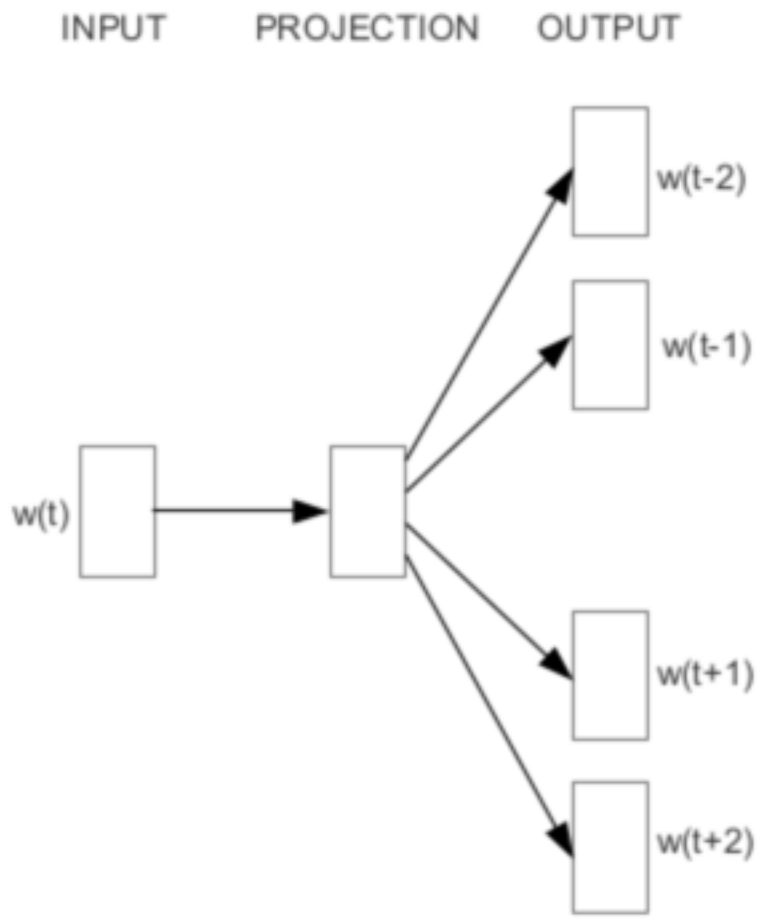
Input: have, big
label: a

I	have	a	big	dog	and	horse

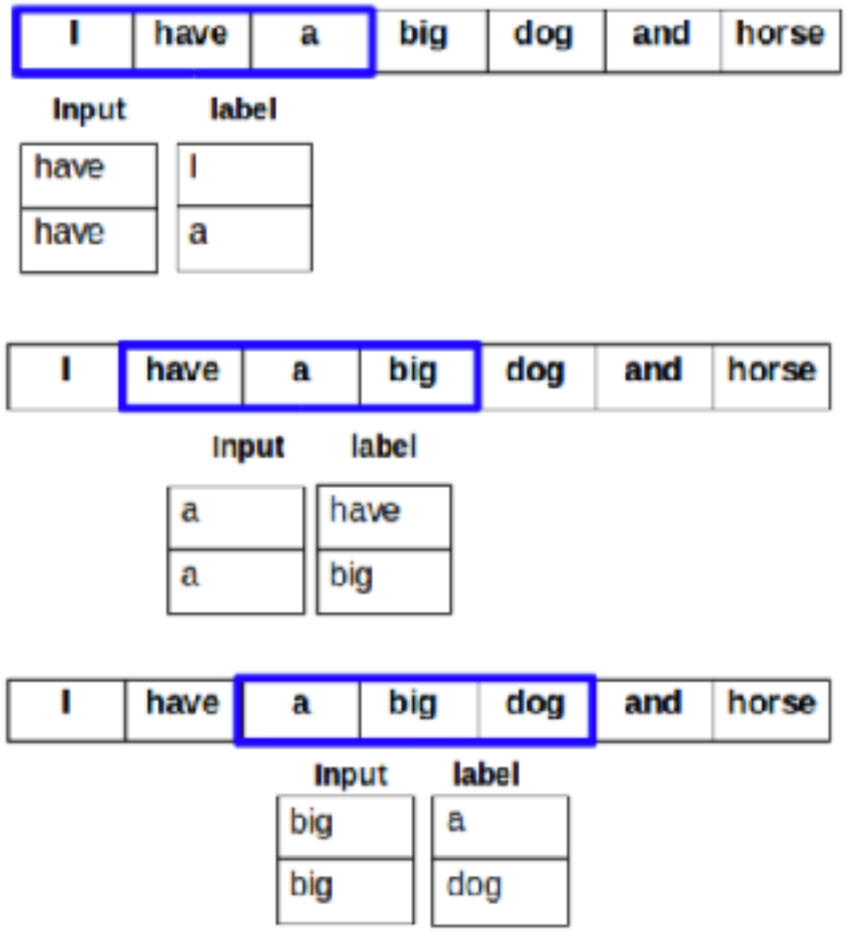
Input: a, dog
label: big

Word2vec

- Skip-gram



Skip-gram



I	have	a	big	dog	and	horse

Input	label
have	I
have	a

I	have	a	big	dog	and	horse

Input	label
a	have
a	big

I	have	a	big	dog	and	horse

Input	label
big	a
big	dog

Thank!

Reference

- http://cpmarkchang.logdown.com
- https://zh.wikipedia.org/wiki/自然语言处理
- http://zake7749.github.io/2016/08/28/word2vec-with-gensim/
- https://www.tensorflow.org/tutorials/word2vec
- http://zongsoftwarenote.blogspot.tw/2017/04/word2vec-model-introduction-skip-gram.html