# Word Embeddings
# Word2Vec

國立臺灣大學 資訊工程學系
陳縕儂 助理教授
http://vivianchen.idv.tw

# Word2Vec – Skip-Gram Model

- Goal: predict surrounding words within a window of each word
- Objective function: maximize the probability of any context word given the current center word

$$w_1, w_2, \cdots, \underbrace{w_{t-m}, \cdots, w_{t-1}}_{w_I}, w_t, \underbrace{w_{t+1}, \cdots, w_{t+m}}_{w_O}, \cdots, w_{T-1}, w_T$$

context window

$$p(w_{O,1}, w_{O,2}, \cdots, w_{O,C} \mid w_I) = \prod_{c=1}^{C} p(w_{O,c} \mid w_I)$$

target word vector

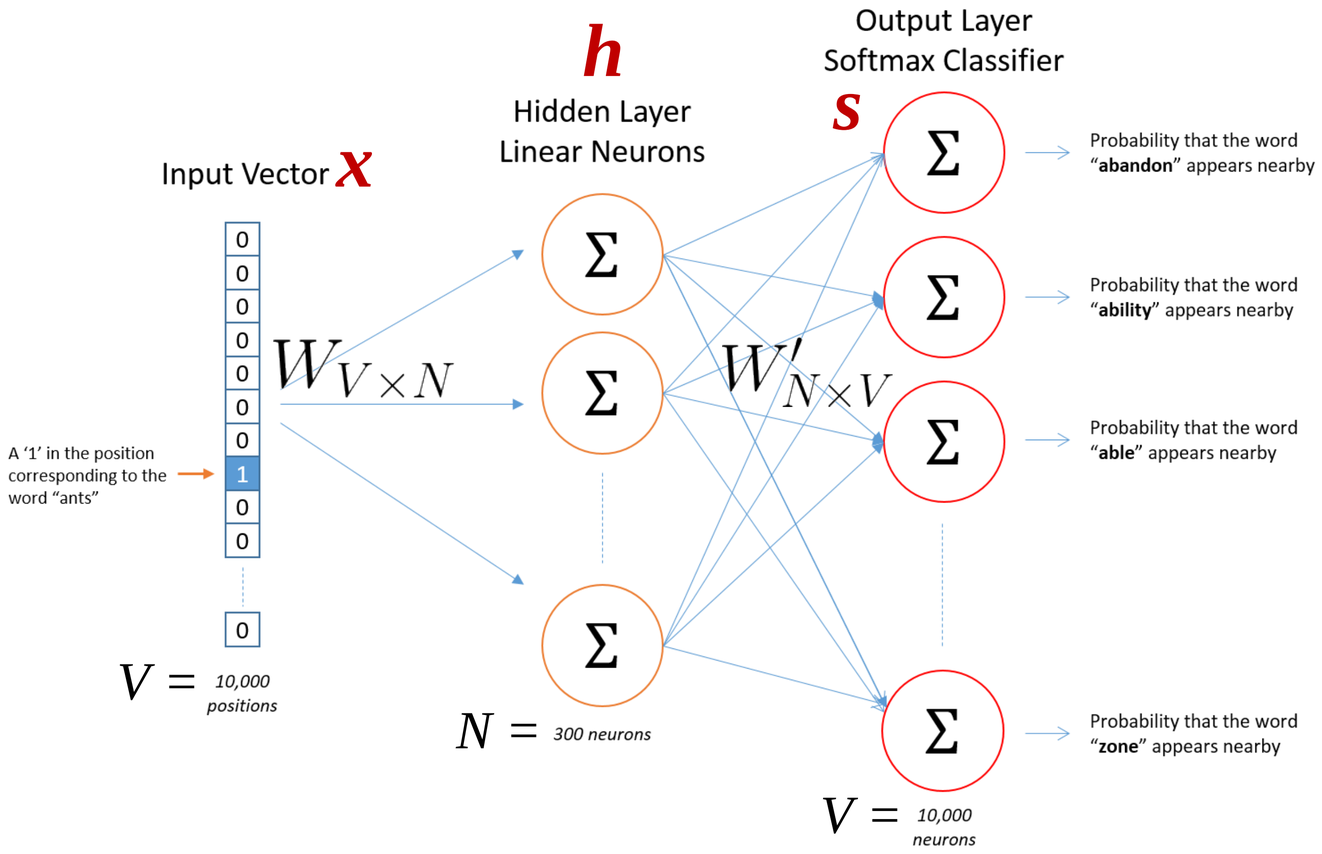
$$C(\theta) = -\sum_{w_I} \sum_{c=1}^{C} \log p(w_{O,c} \mid w_I) \quad p(w_O \mid w_I) = \frac{\exp(v'^T_{w_O} v_{w_I})}{\sum_j \exp(v'^T_{w_j} v_{w_I})}$$

outside target word

Benefit: faster, easily incorporate a new sentence/document or add a word to vocab

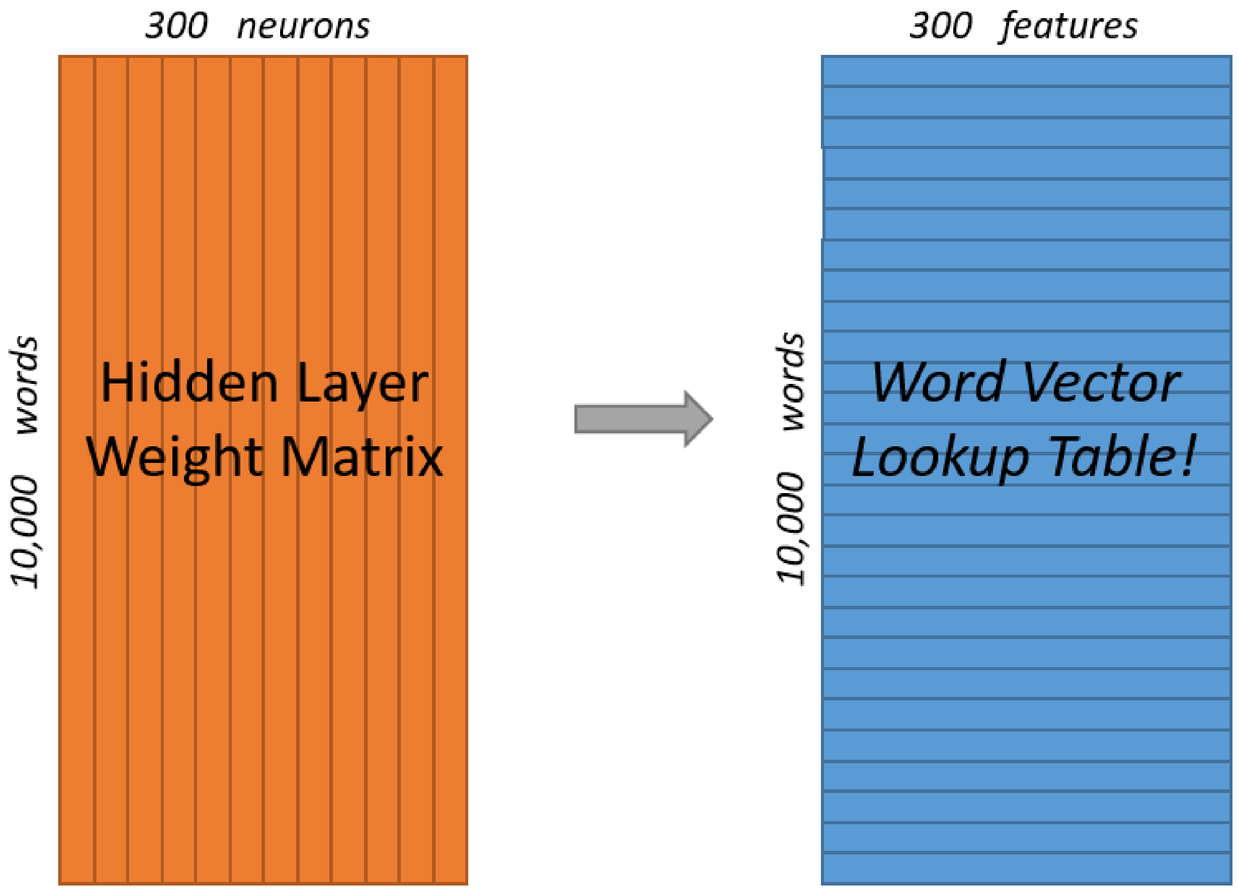
# Word2Vec Skip-Gram Illustration

- Goal: predict surrounding words within a window of each word




# Hidden Layer Matrix → Word Embedding Matrix

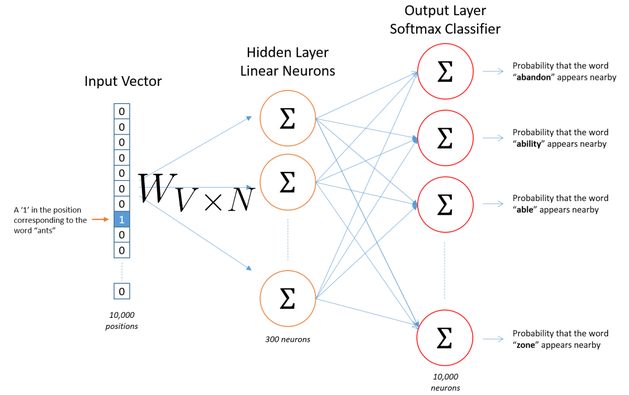
$$W_{V \times N}$$



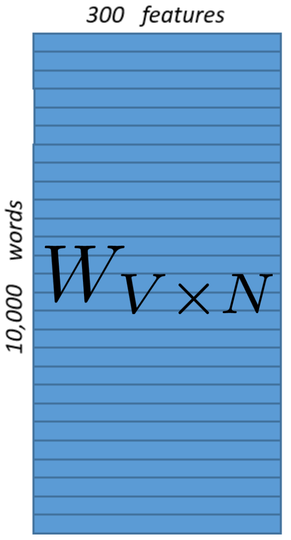

# Weight Matrix Relation

- Hidden layer weight matrix = word vector lookup

$$h = x^T W = W_{(k,.)} := v_{w_I}$$



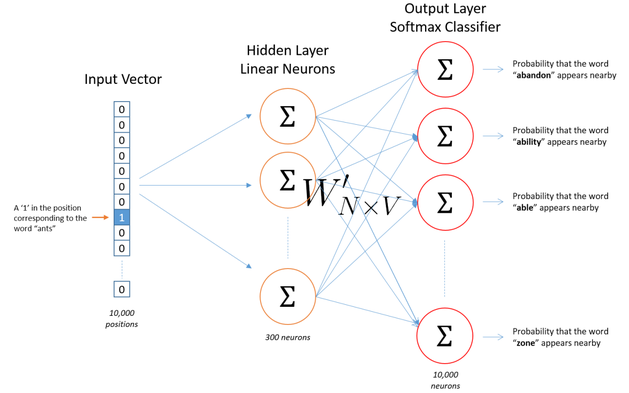

$$[0 \quad 0 \quad 0 \quad 1 \quad 0] \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} = [10 \quad 12 \quad 19]$$



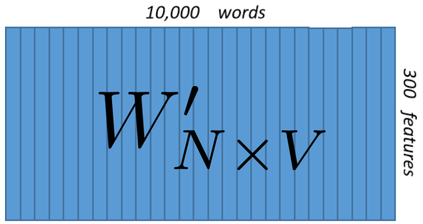

Each vocabulary entry has two vectors: as a **target** word and as a **context** word

# Weight Matrix Relation

- Output layer weight matrix = weighted sum as final score




$$s_j = hv'_{w_j}$$

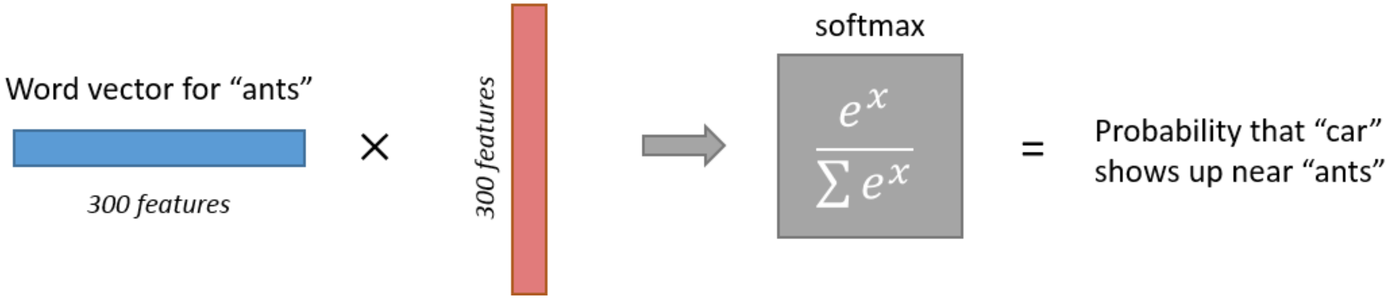



$$p(w_j = w_{O,c} \mid w_I) = y_{j_c} = \frac{\exp(s_{j_c})}{\sum_{j'=1}^{V} \exp(s_{j'})}$$

softmax

within the context window

Output weights for "car"




Each vocabulary entry has two vectors: as a **target** word and as a **context** word

# Word2Vec Skip-Gram Illustration