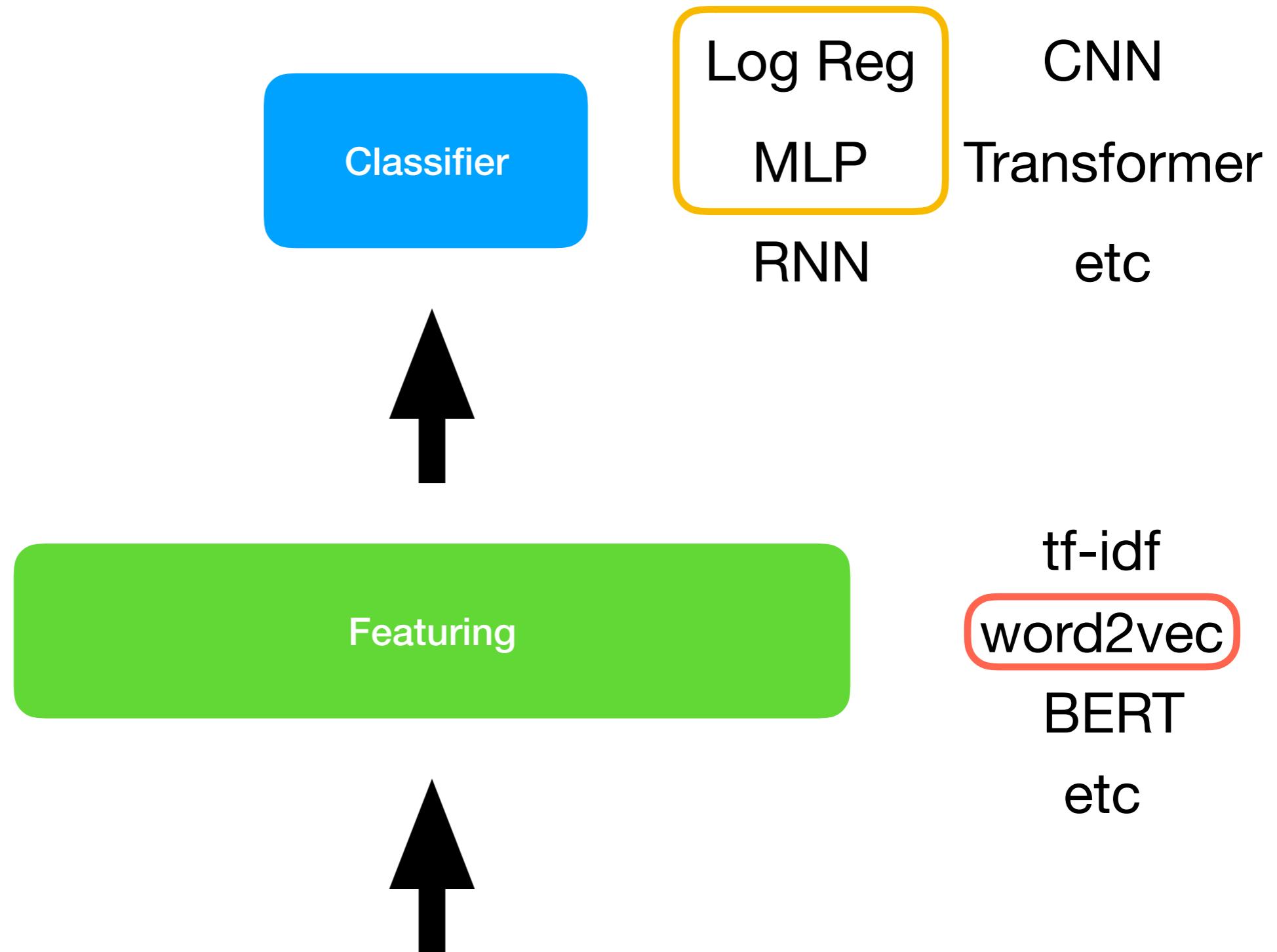


# Word Embeddings

# Training Pipeline



The iPhone X is the huge leap forward

# One Hot Encoding

Bag of words

```
motel = [0 0 0 0 0 0 0 0 0 1 0 0 0]  
hotel = [0 0 0 0 0 0 1 0 0 0 0 0 0]
```

**Ortogonal vectors**

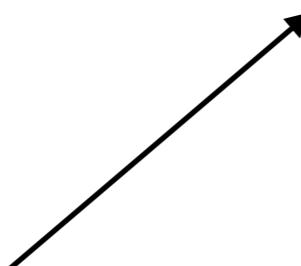
**Dimension = len(vocabulary)**

# Similarity

Dot Product

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

[0, 1]



Vector Norms

# TF-IDF

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

**TF-IDF**

Term  $x$  within document  $y$

$tf_{x,y}$  = frequency of  $x$  in  $y$

$df_x$  = number of documents containing  $x$

$N$  = total number of documents

# TF-IDF

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

**TF-IDF**

Term  $x$  within document  $y$

$tf_{x,y}$  = frequency of  $x$  in  $y$

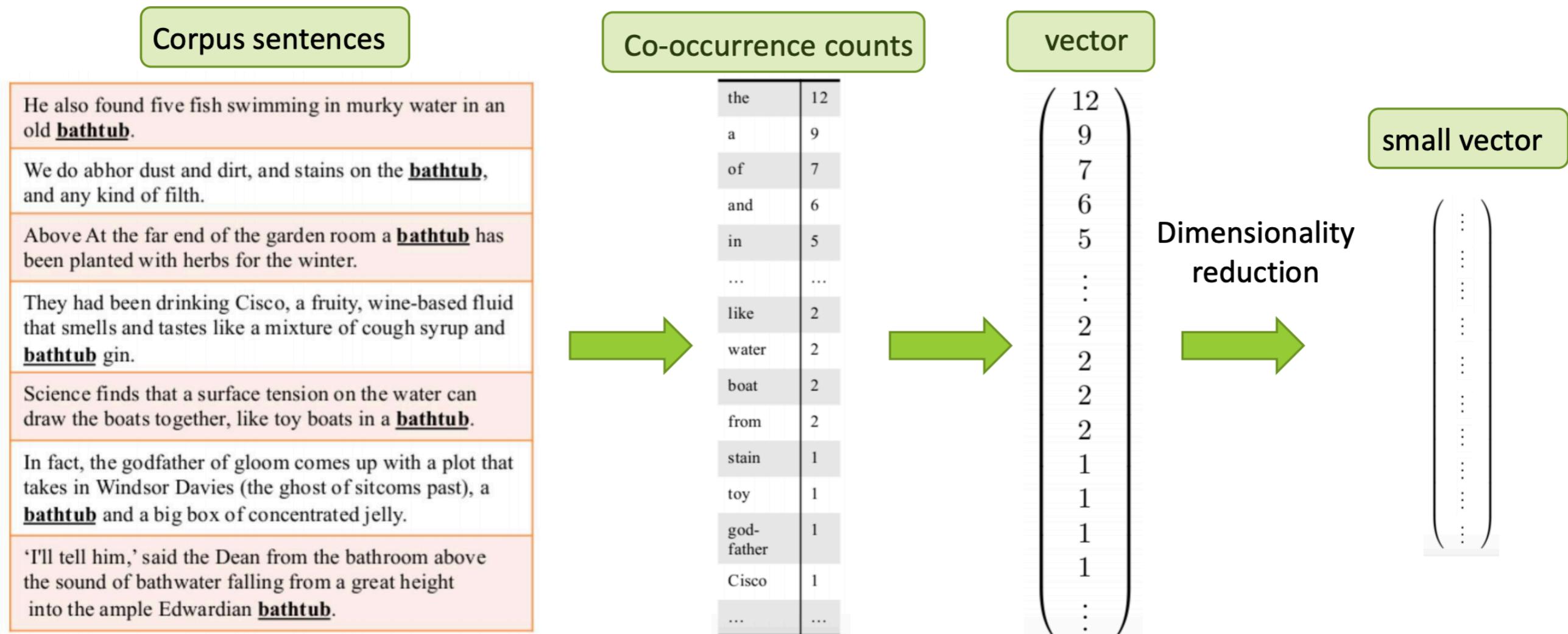
$df_x$  = number of documents containing  $x$

$N$  = total number of documents

<b>text1</b>	0	0	0	0	0.47	0	0.23	0
<b>text2</b>	0	0.68	0	0	0.32	0	0	0
<b>text3</b>	0.11	0	0.19	0	0	0	0	0

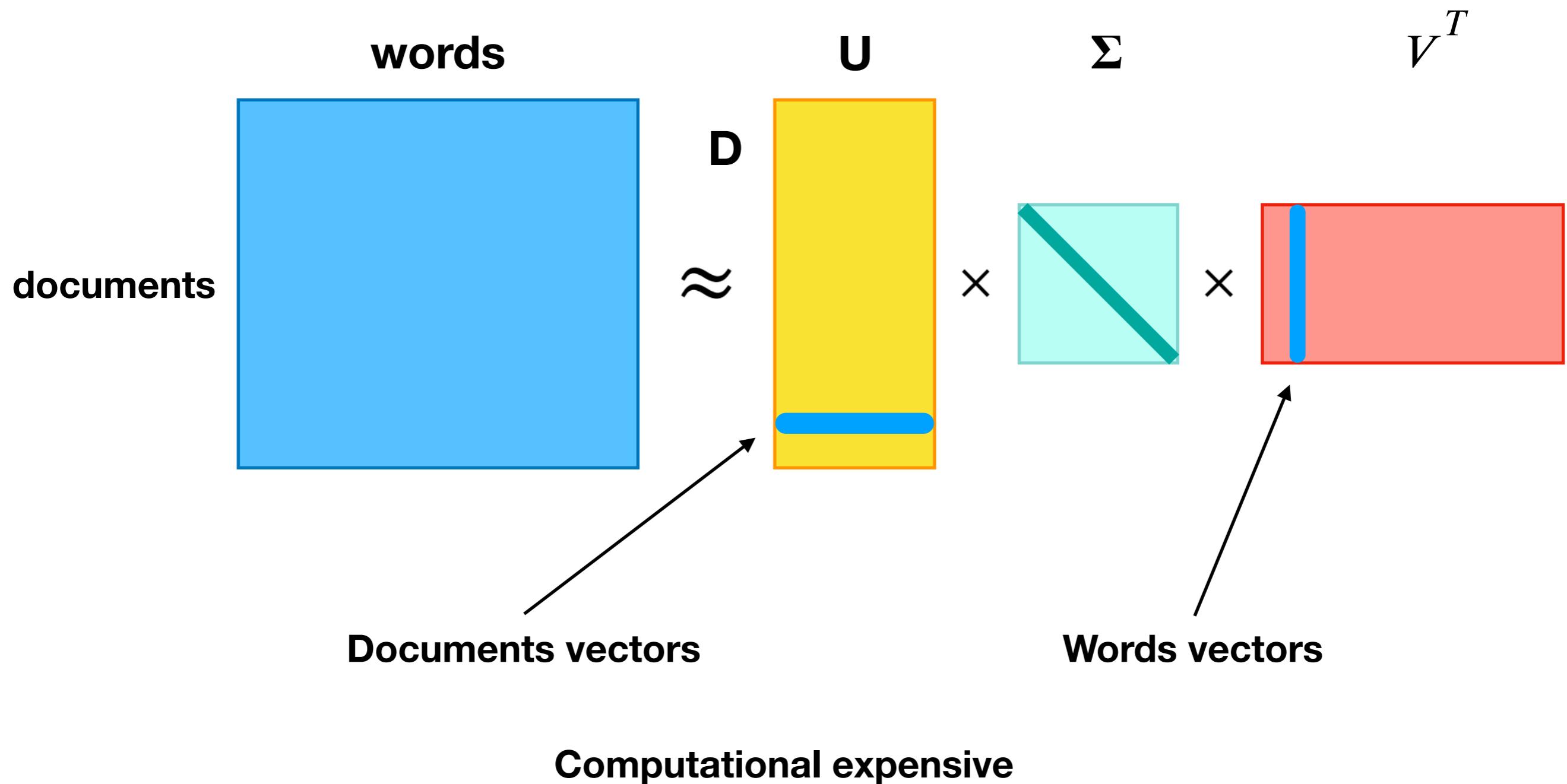
# Co-occurrence Vectors

«*You shall know a word by the company it keeps*» – Firth, 1957



# Co-occurrence Matrix

$$X \approx \hat{X} = U \Sigma V^T$$



# Co-occurrence Matrix

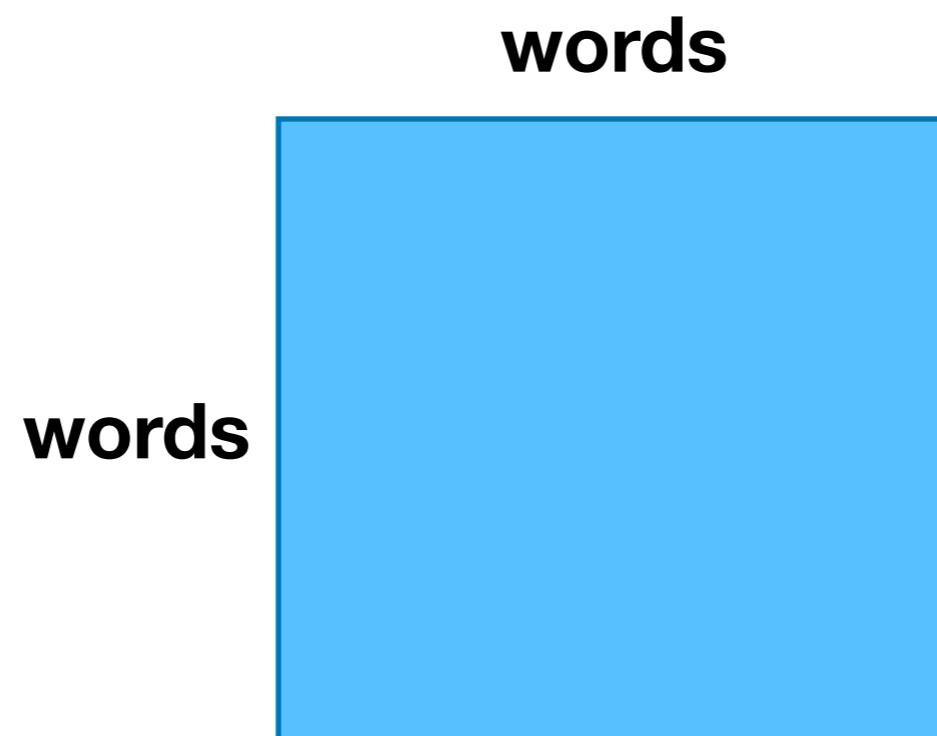
words



words

№	Словосочетание	Документы	Частота
1	<u>и не</u>	22732	201352
2	<u>и в</u>	27048	193983
3	<u>потому что</u>	14926	117401
4	<u>я не</u>	10675	113767
5	<u>у меня</u>	9734	97102
6	<u>может быть</u>	16086	96065
7	<u>то что</u>	17195	95251
8	<u>что он</u>	11786	92743
9	<u>не было</u>	13196	92729
10	<u>в том</u>	21604	89842

# Co-occurrence Matrix

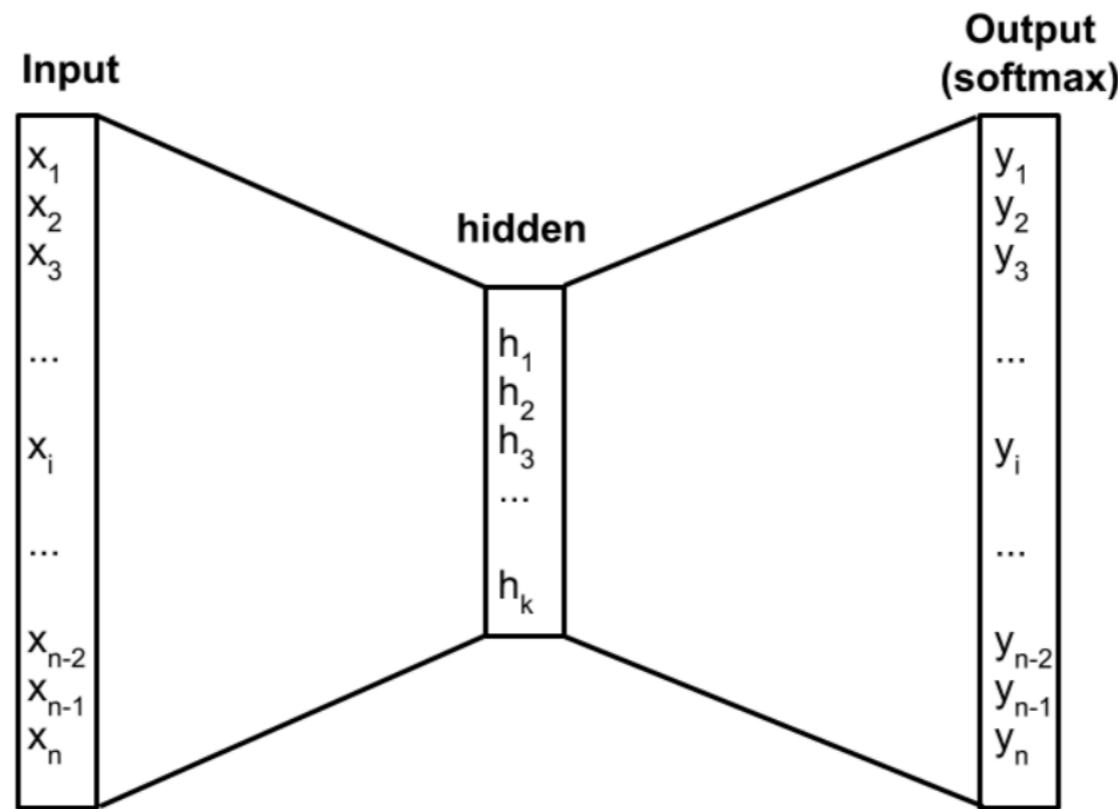


$$\text{pmi}(x; y) \equiv \log \frac{p(x, y)}{p(x)p(y)} = \log \frac{p(x|y)}{p(x)} = \log \frac{p(y|x)}{p(y)}$$

$$\text{ppmi} = \max(\text{pmi}, 0)$$

# Word2Vec

# Word2Vec



$$\frac{1}{T} \sum_{t=1}^T \sum_{-c \leq j \leq c, j \neq 0} \log p(w_{t+j} | w_t)$$

# Word2Vec

## Source Text

The quick brown fox jumps over the lazy dog. →

The quick brown fox jumps over the lazy dog. →

The quick brown fox jumps over the lazy dog. →

The quick brown fox jumps over the lazy dog. →

## Training Samples

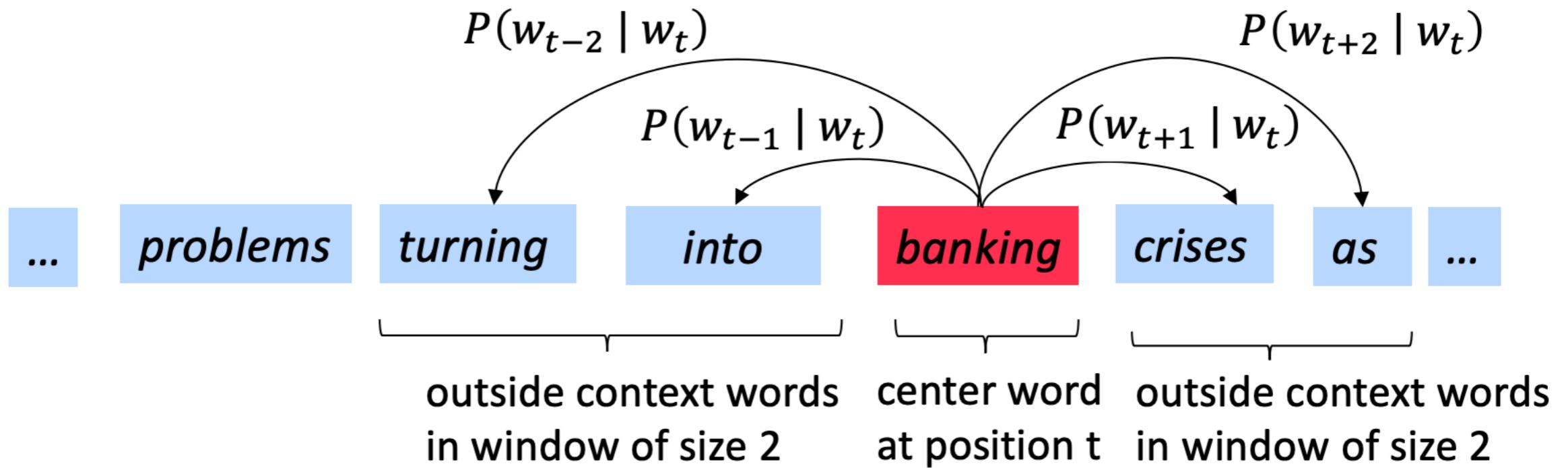
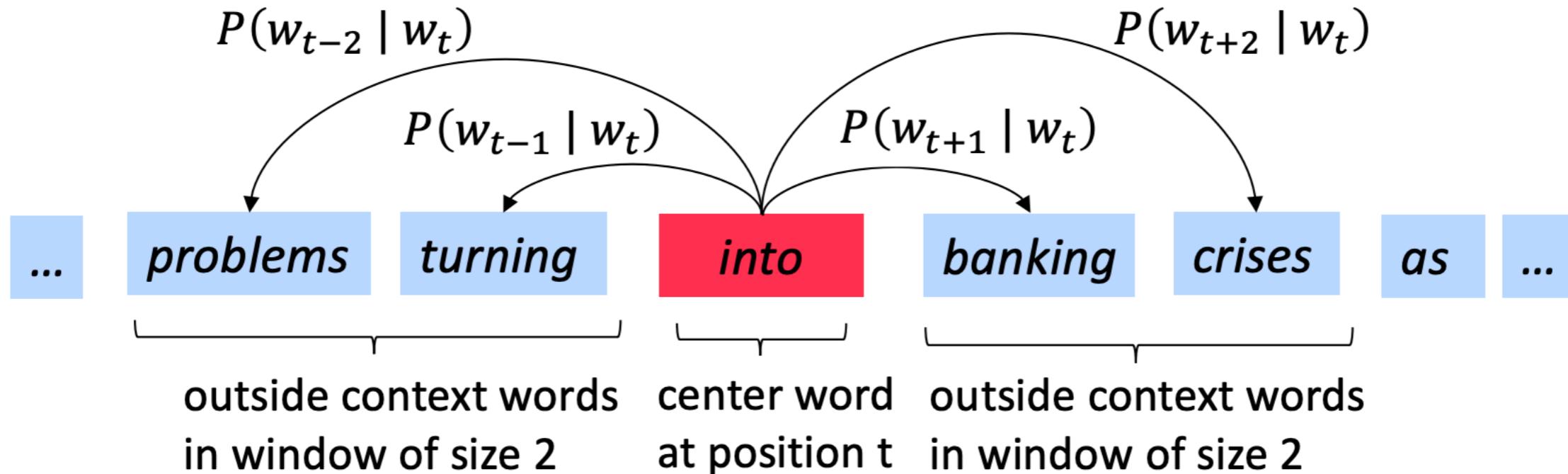
(the, quick)  
(the, brown)

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

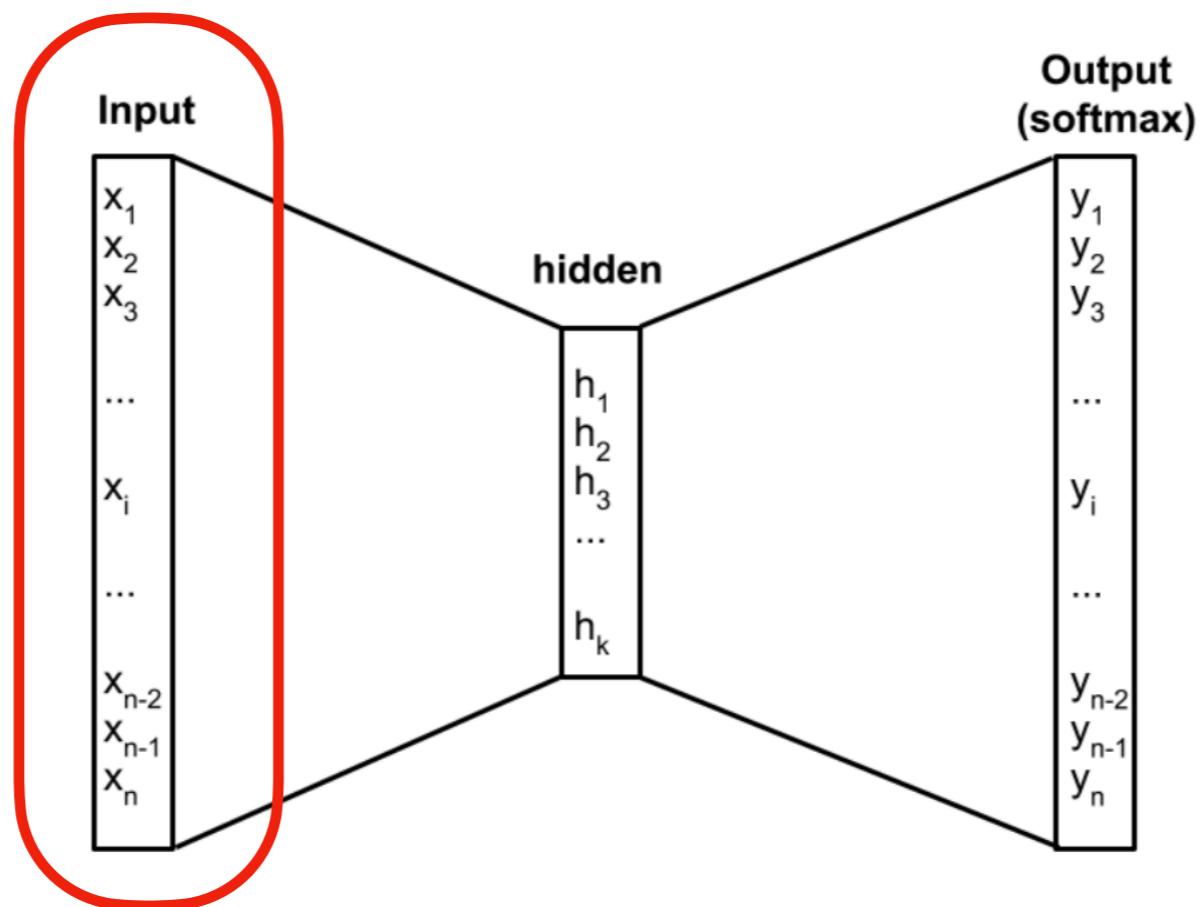
(brown, the)  
(brown, quick)  
(brown, fox)  
(brown, jumps)

(fox, quick)  
(fox, brown)  
(fox, jumps)  
(fox, over)

# Word2Vec

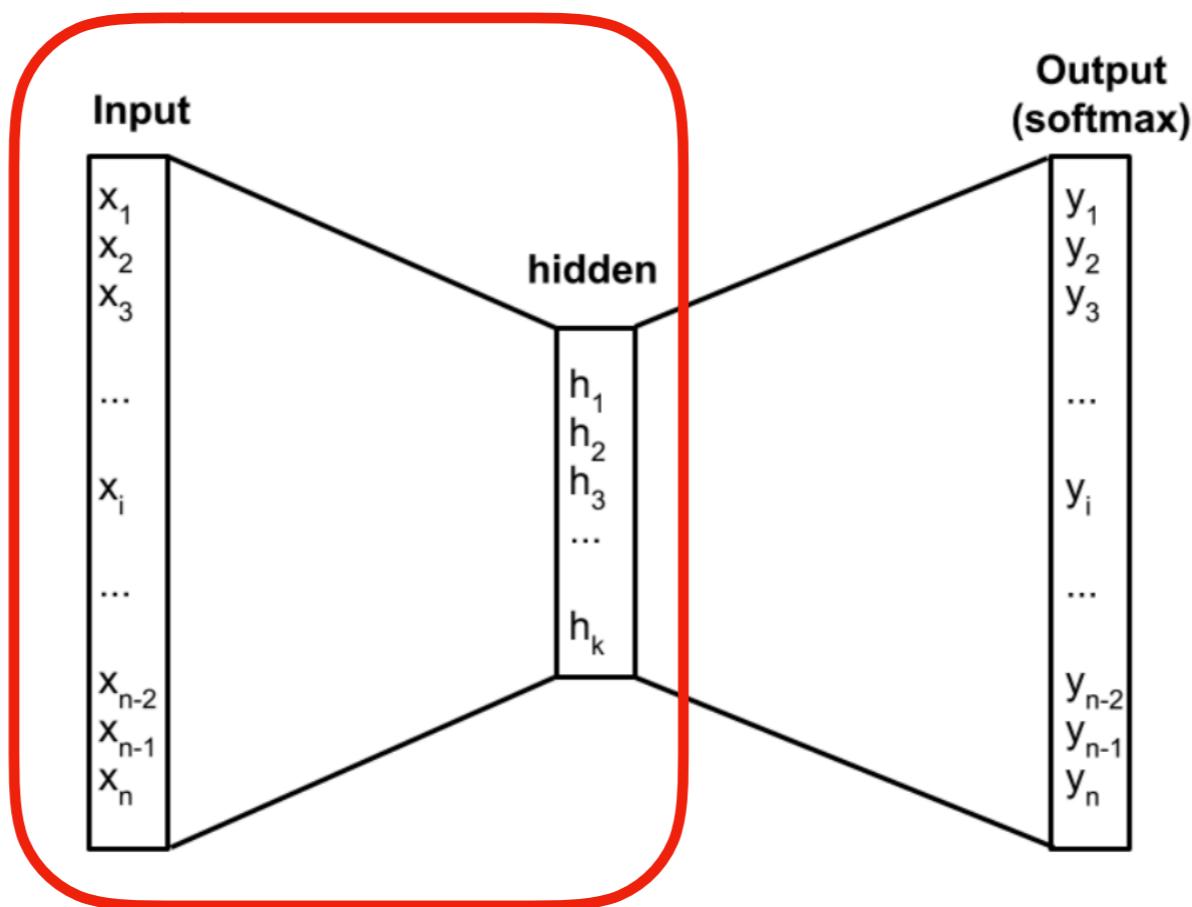


# Word2Vec



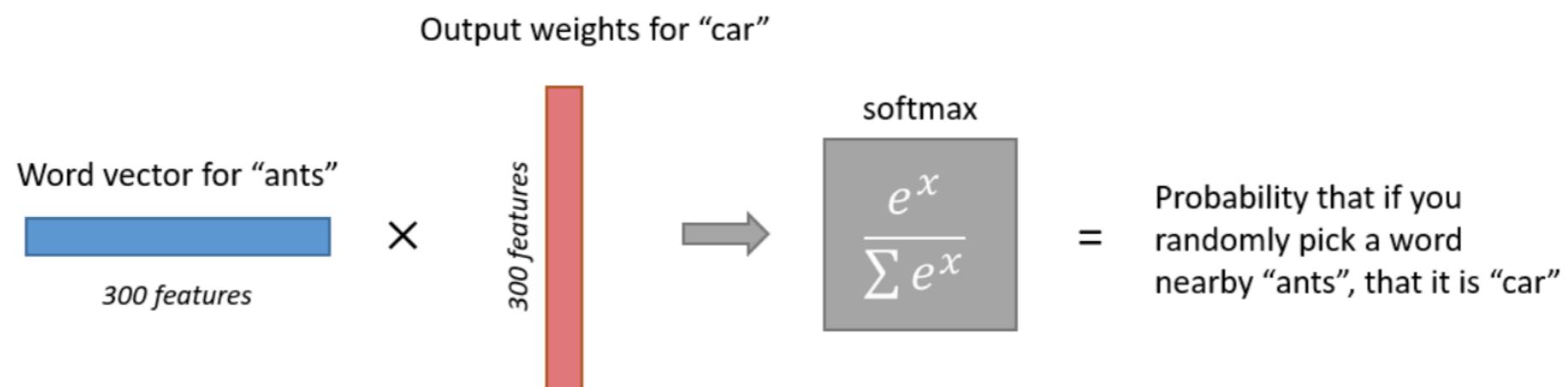
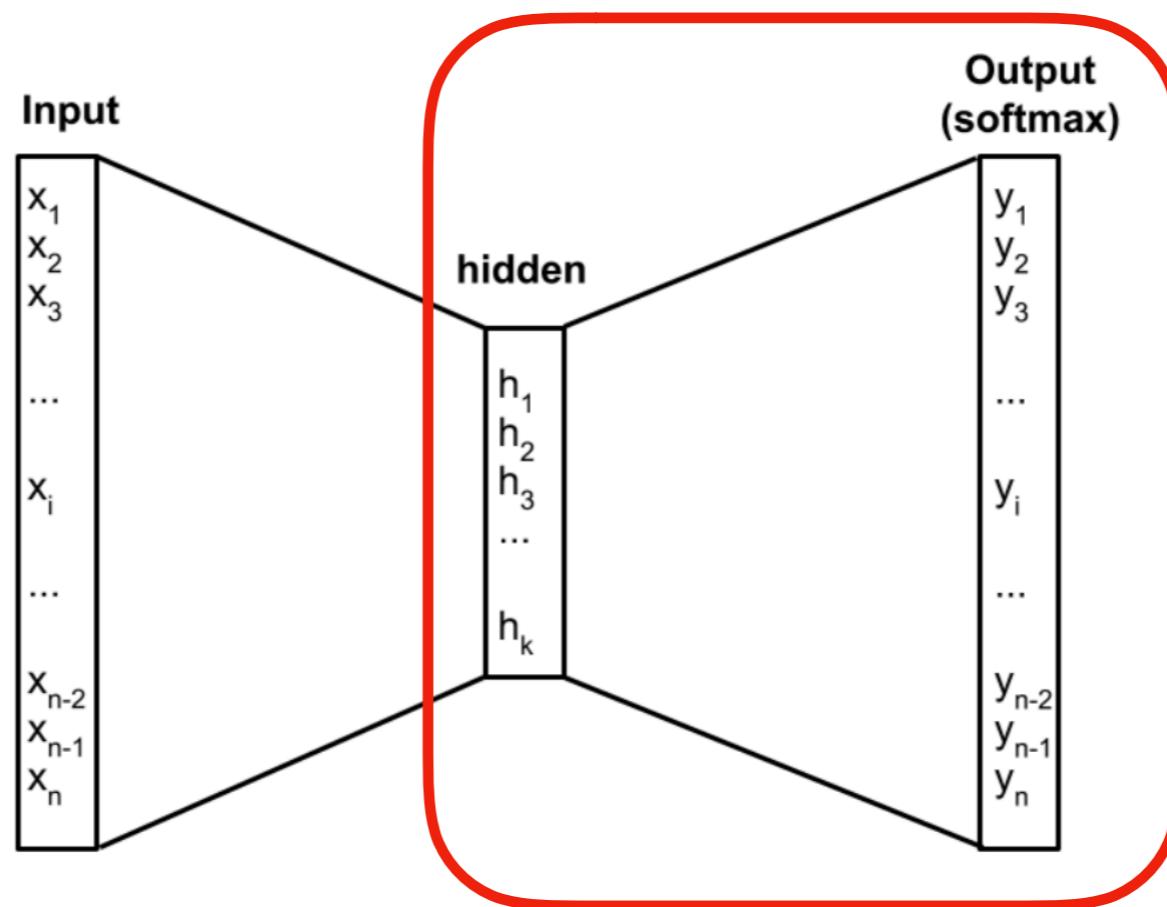
$$\theta = \begin{bmatrix} v_{aardvark} \\ v_a \\ \vdots \\ v_{zebra} \\ u_{aardvark} \\ u_a \\ \vdots \\ u_{zebra} \end{bmatrix} \in \mathbb{R}^{2dV}$$

# Word2Vec

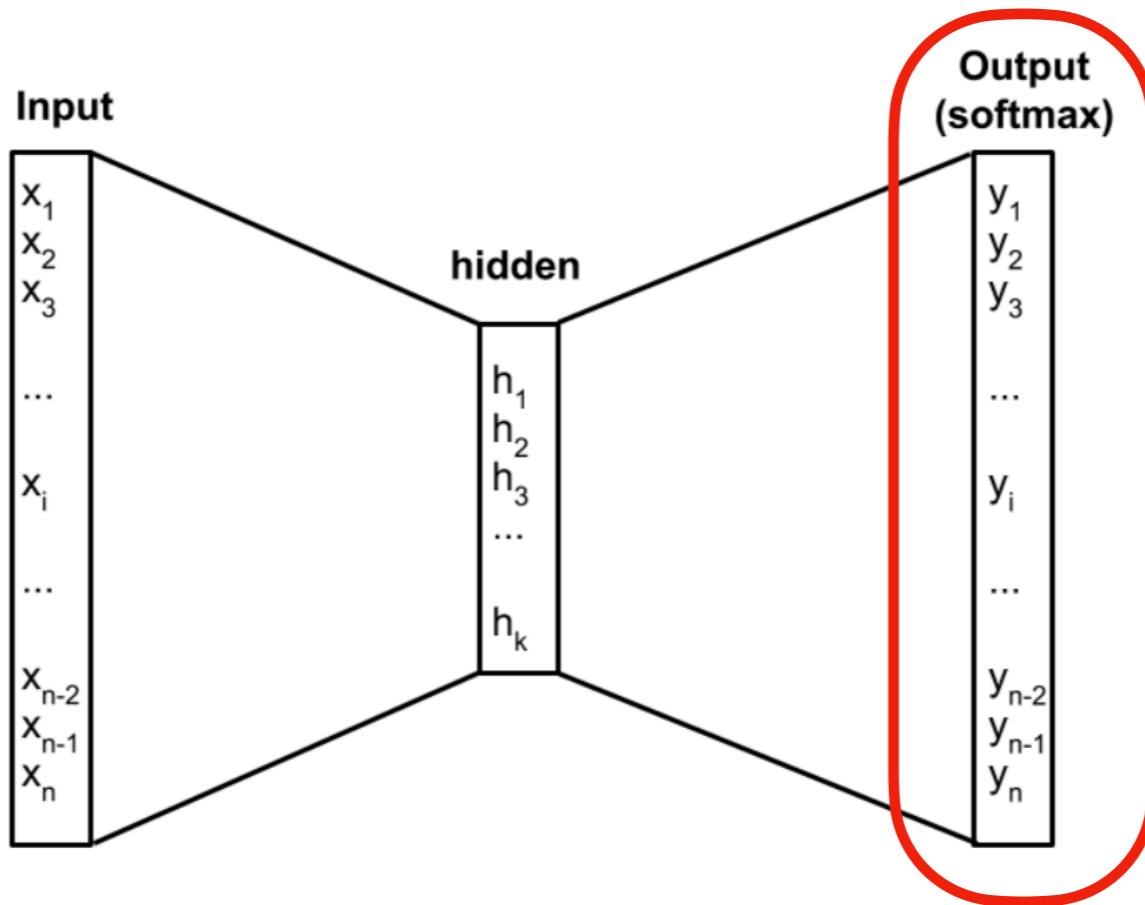


$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} = [10 \quad 12 \quad 19]$$

# Word2Vec

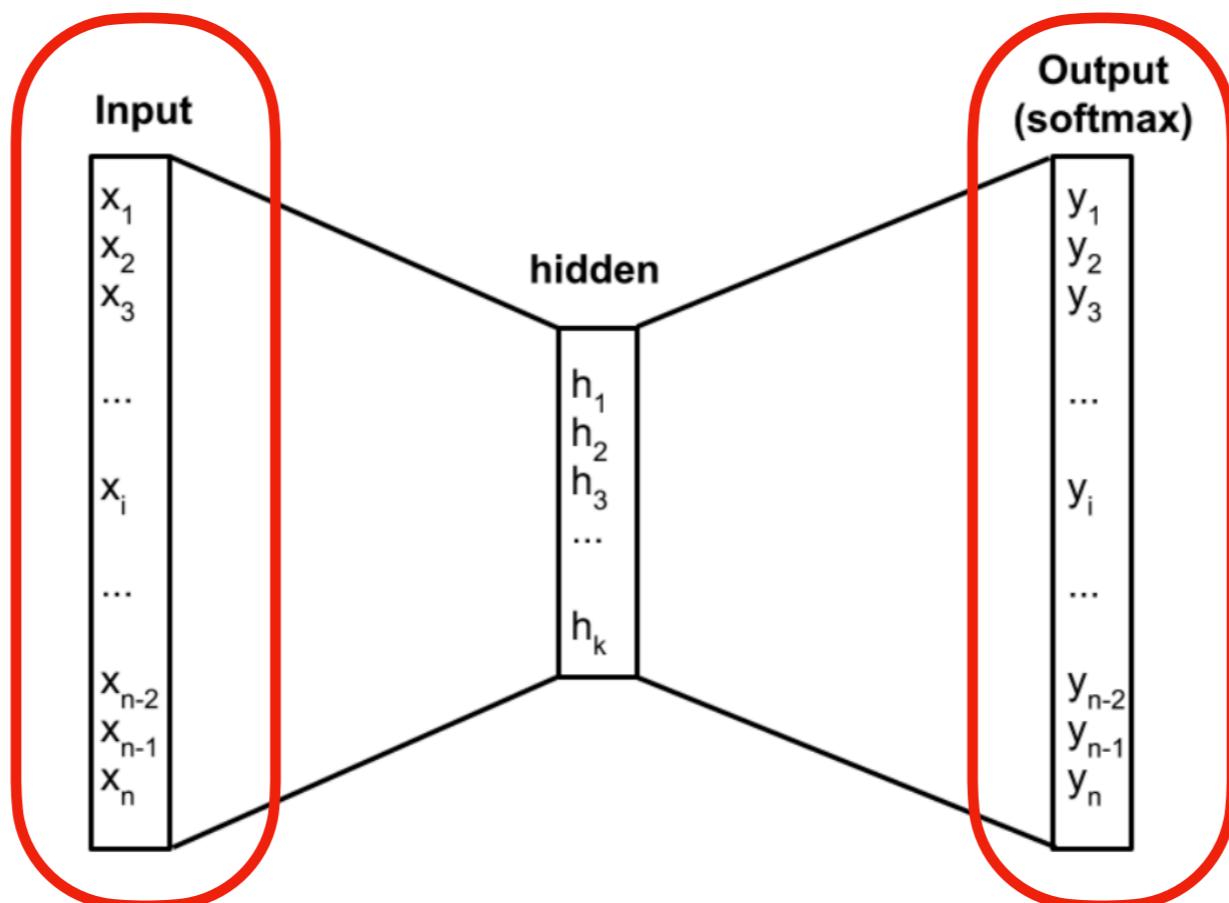


# Word2Vec



$$p(w_O | w_I) = \frac{\exp(v'_{w_O}^\top v_{w_I})}{\sum_{w=1}^W \exp(v'_{w'}^\top v_{w_I})}$$

# Word2Vec



$$\theta = \begin{bmatrix} v_{aardvark} \\ v_a \\ \vdots \\ v_{zebra} \\ u_{aardvark} \\ u_a \\ \vdots \\ u_{zebra} \end{bmatrix} \in \mathbb{R}^{2dV}$$

**Same words**

$$\theta = \begin{bmatrix} v_{aardvark} \\ v_a \\ \vdots \\ v_{zebra} \\ u_{aardvark} \\ u_a \\ \vdots \\ u_{zebra} \end{bmatrix} \in \mathbb{R}^{2dV}$$

**Different vectors**

# Word2Vec

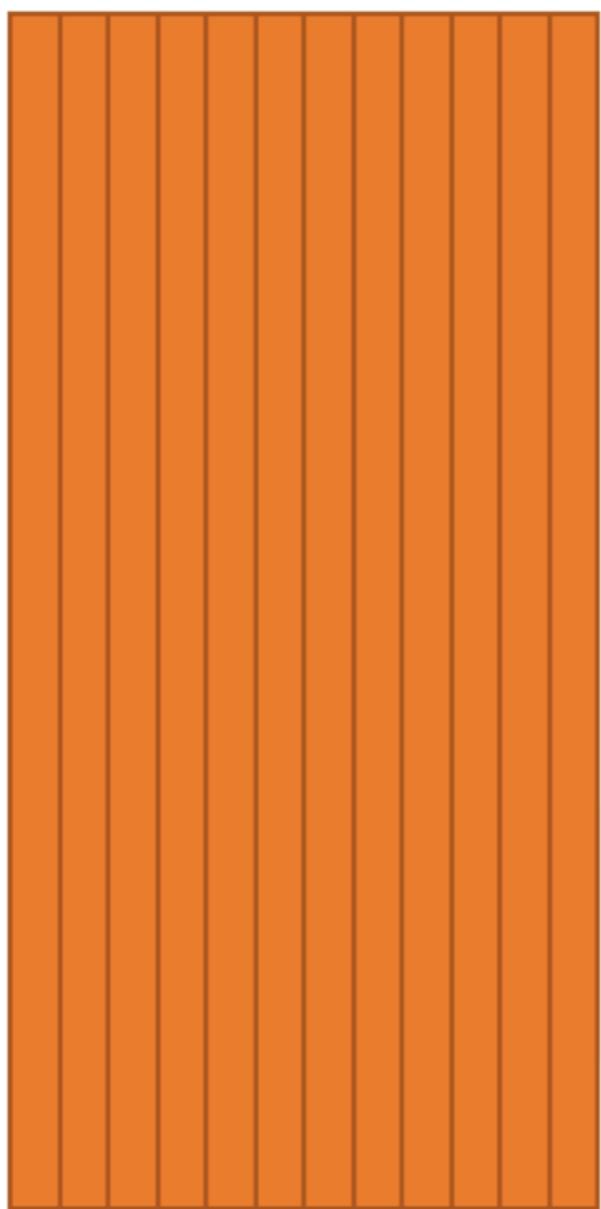
Hidden Layer  
Weight Matrix



*Word Vector  
Lookup Table!*

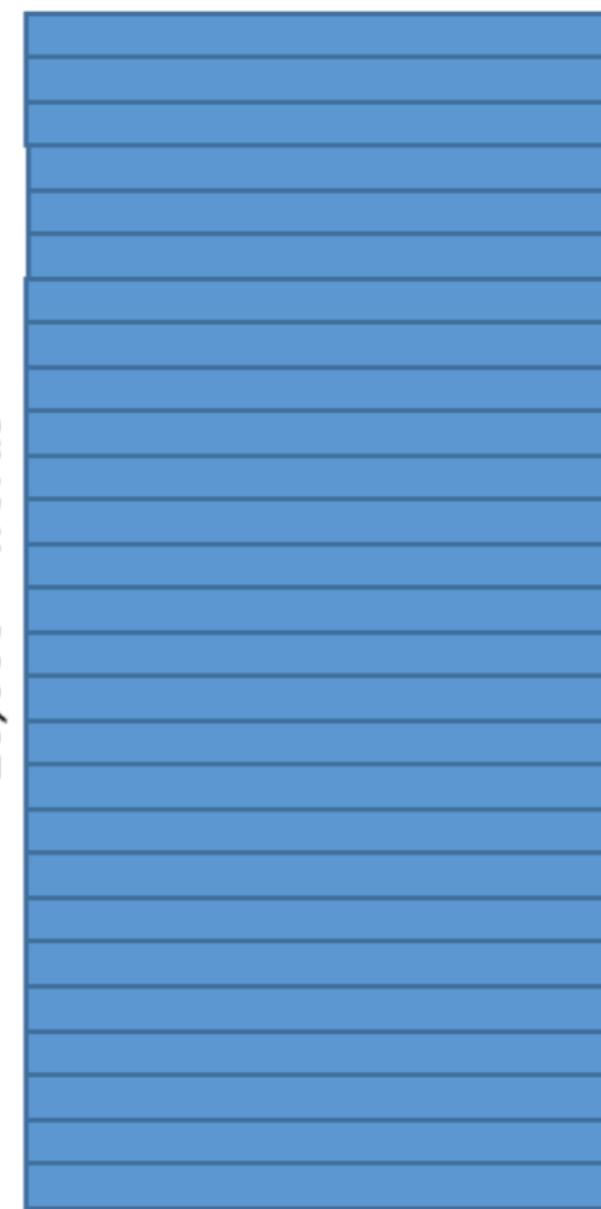
*300 neurons*

*10,000 words*

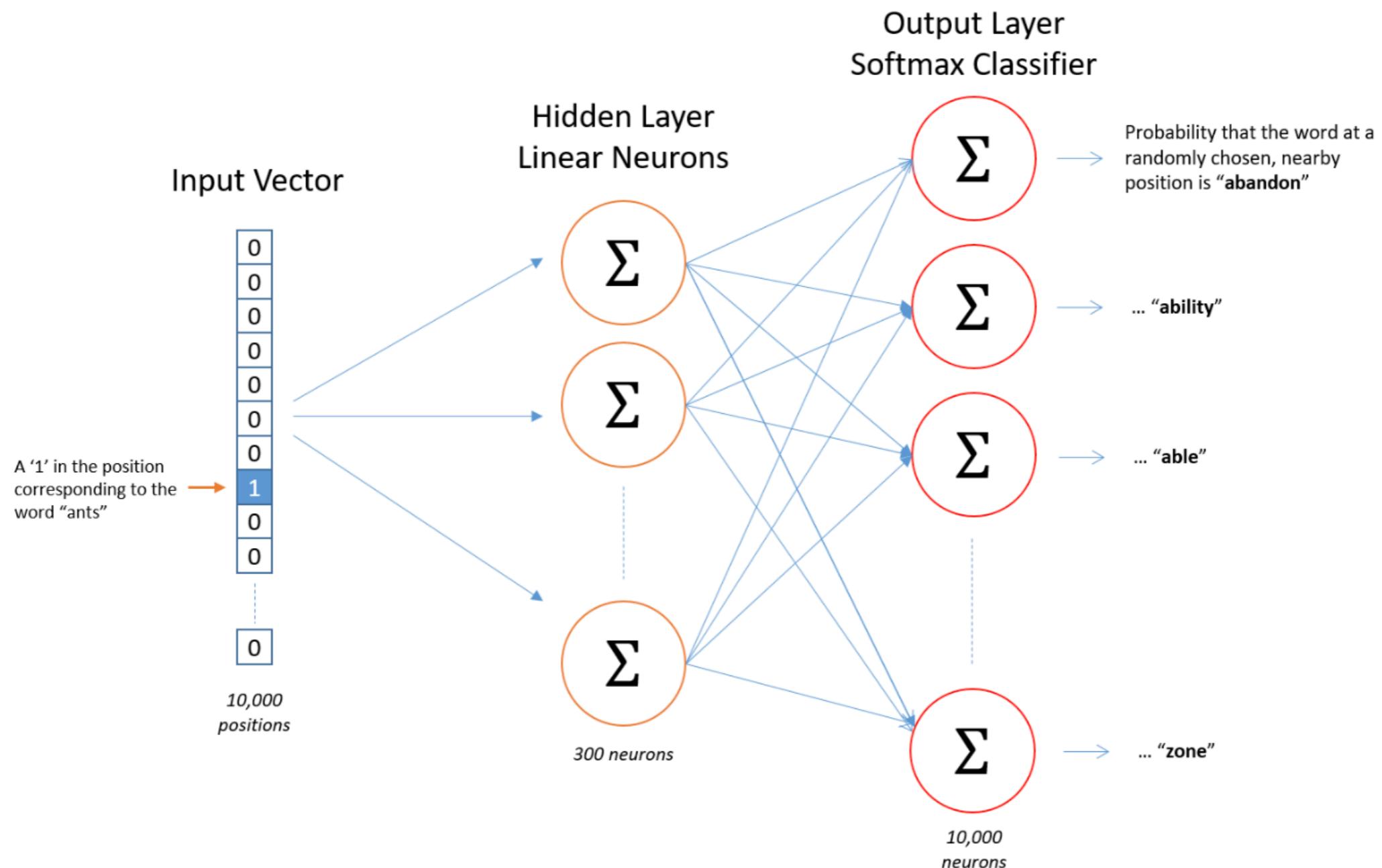


*300 features*

*10,000 words*



# Word2Vec

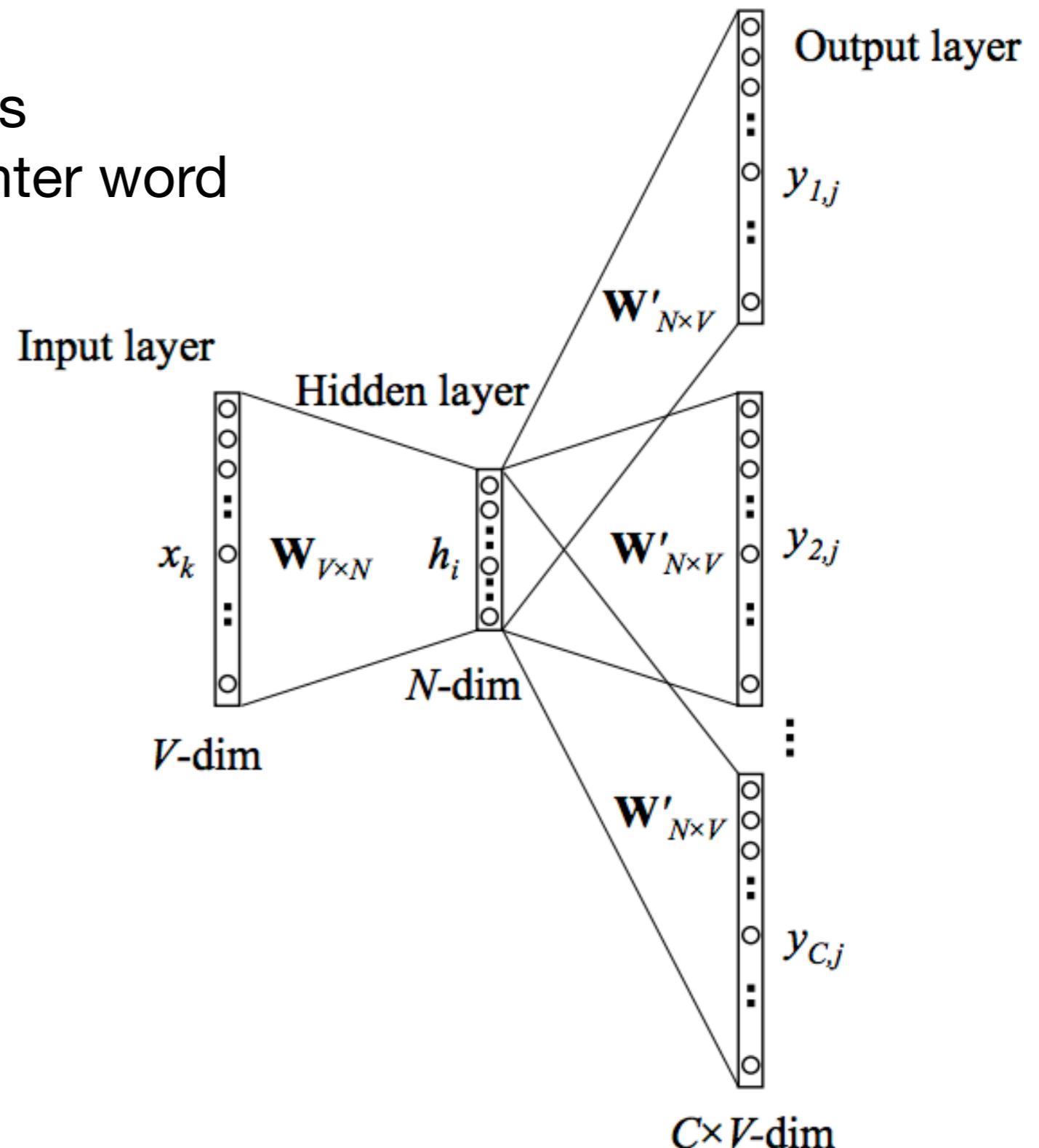


# Word2Vec

## Skipgrams

Predict context ("outside") words  
(position independent) given center word

Better for rare words

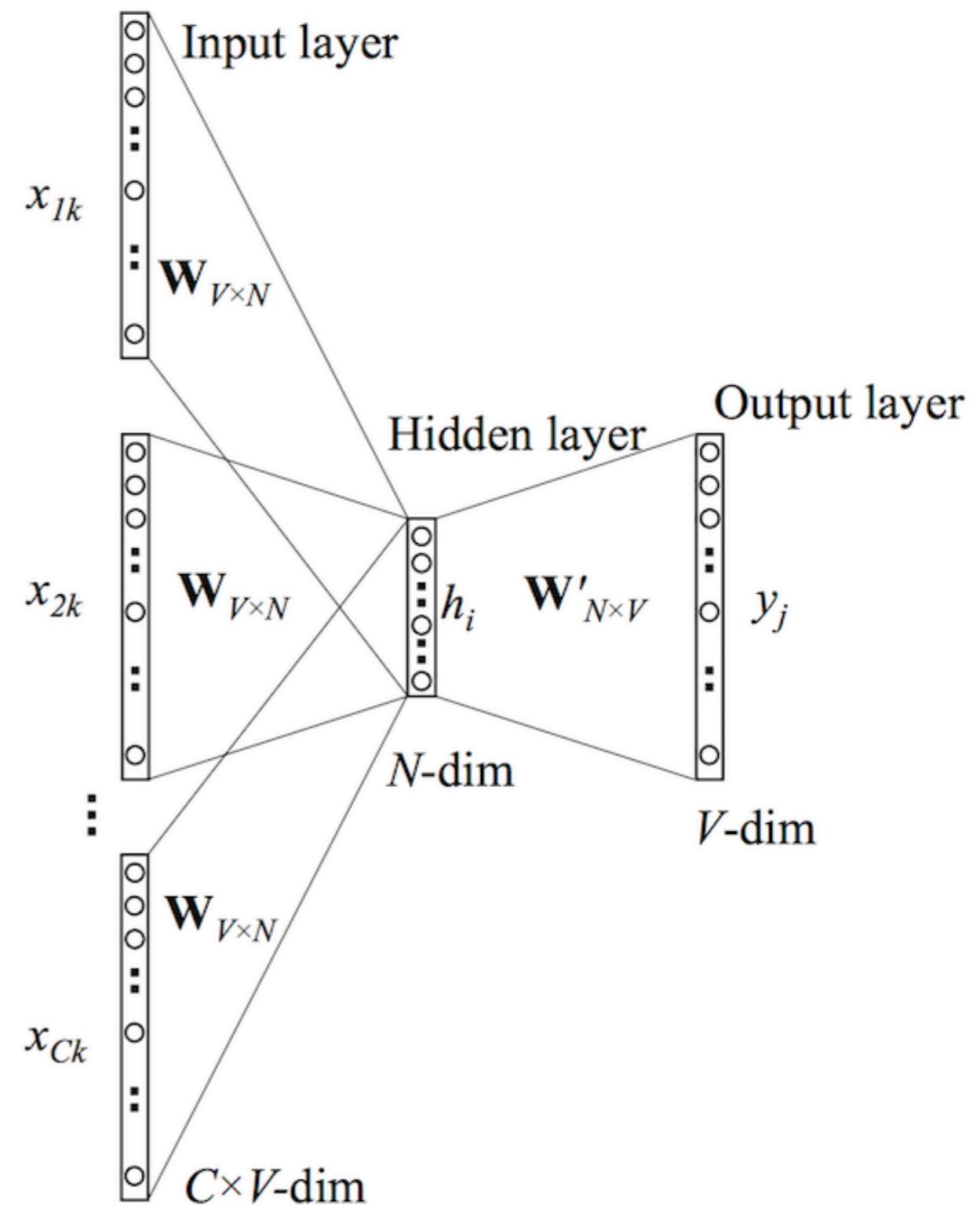


# Word2Vec

CBOW

Predict center word from  
(bag of) context words

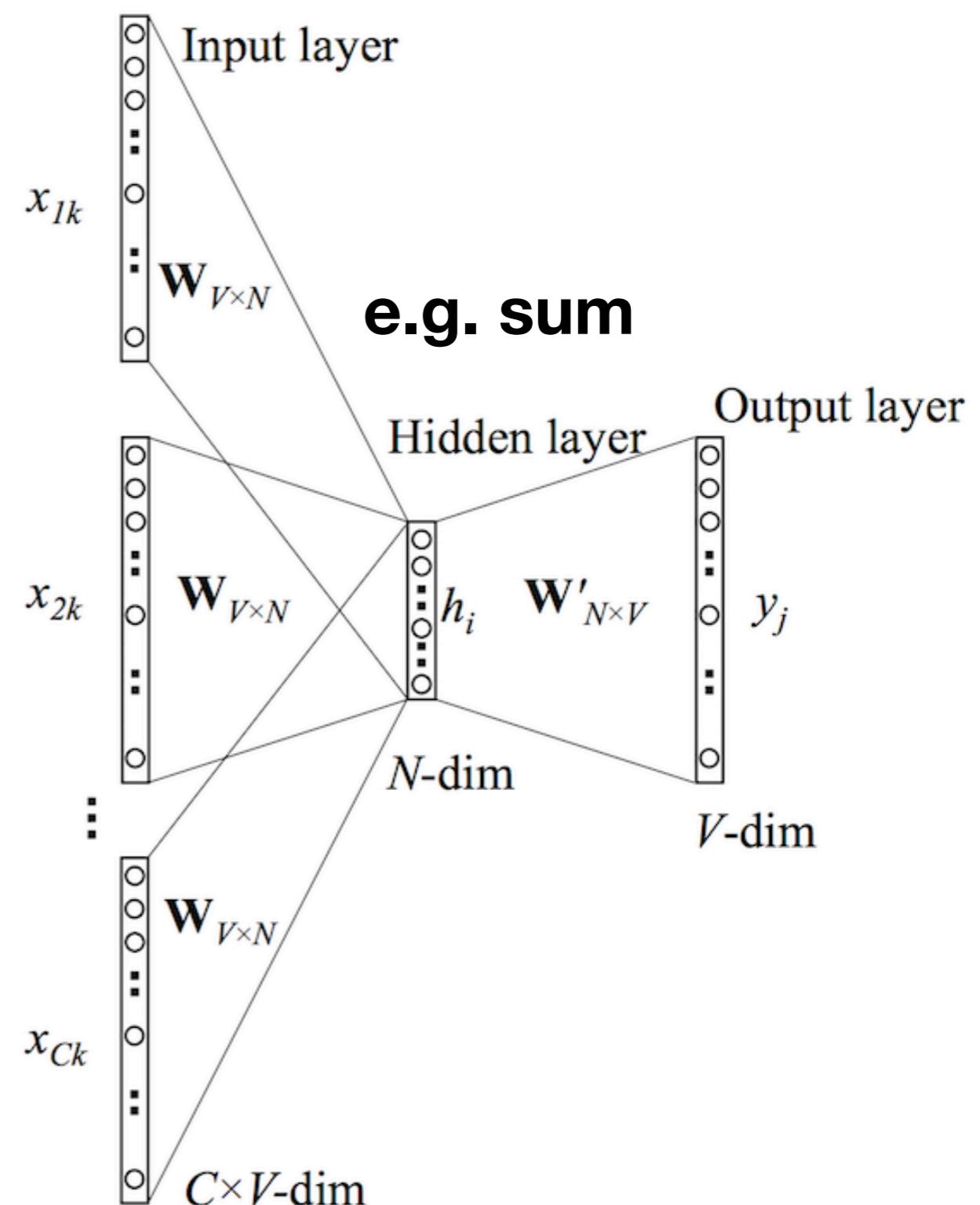
Faster



# Word2Vec

CBOW

Predict center word from  
(bag of) context words

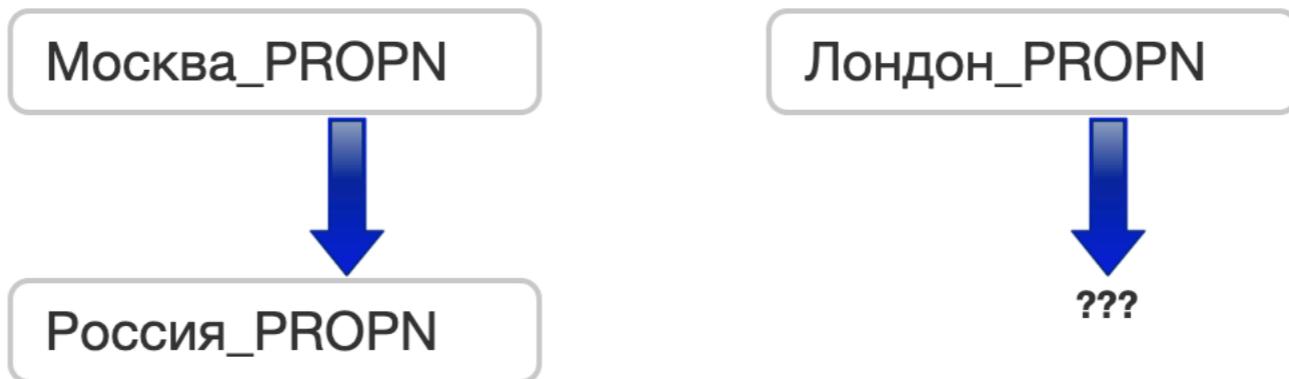


# Visualization

<https://projector.tensorflow.org/>

- BERT Embedding Projector

# Visualization



Частотность слова

Высокая  Средняя  Низкая

## НКРЯ и Wikipedia

- |    |                |                       |      |  |
|----|----------------|-----------------------|------|--|
| 1. | англия         | <small>PROP_N</small> | 0.58 |  |
| 2. | европа         | <small>PROP_N</small> | 0.54 |  |
| 3. | великобритания | <small>PROP_N</small> | 0.52 |  |
| 4. | страна         | <small>NOUN</small>   | 0.48 |  |
| 5. | франция        | <small>PROP_N</small> | 0.47 |  |

# Visualization

**Some vector close to queen**

**word2vec(king) - word2vec(man) + word2vec(woman) = word2vec (queen)**

# Word2Vec

Fasttext

OOV

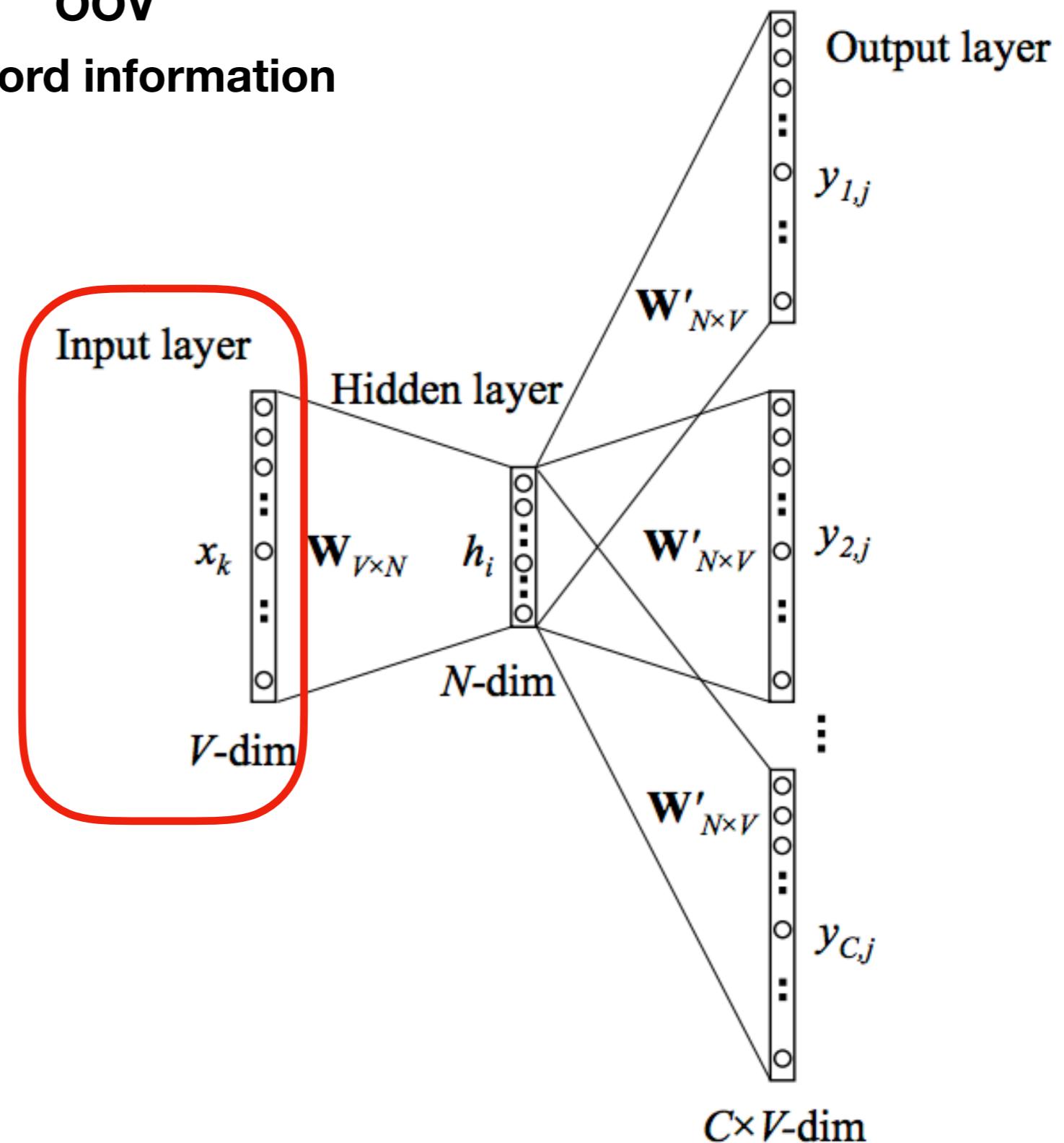
Subword information

where

=

**<wh + whe + her + ere + re>**

3 – 6 char n-gram length

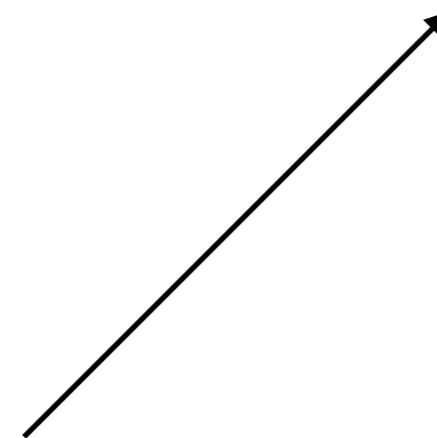


# Word2Vec

$$p(w_O | w_I) = \frac{\exp\left({v'_{w_O}}^\top v_{w_I}\right)}{\sum_{w=1}^W \exp\left({v'_{w}}^\top v_{w_I}\right)}$$

# Word2Vec

$$p(w_O | w_I) = \frac{\exp(v'_{w_O}^\top v_{w_I})}{\sum_{w=1}^W \exp(v'_{w}^\top v_{w_I})}$$



**Computational expensive**

# Word2Vec

- Hierarchical softmax
- Naive softmax
  - Subset of vocabulary
- Negative sampling
  - Binary classification

$$p(w_O | w_I) = \frac{\exp(v'_{w_O}^\top v_{w_I})}{\sum_{w=1}^W \exp(v'_{w}^\top v_{w_I})}$$

# Word2Vec

## Negative sampling

$$J_{\text{neg-sample}}(\mathbf{o}, \mathbf{v}_c, \mathbf{U}) = -\log(\sigma(\mathbf{u}_o^\top \mathbf{v}_c)) - \sum_{k=1}^K \log(\sigma(-\mathbf{u}_k^\top \mathbf{v}_c))$$

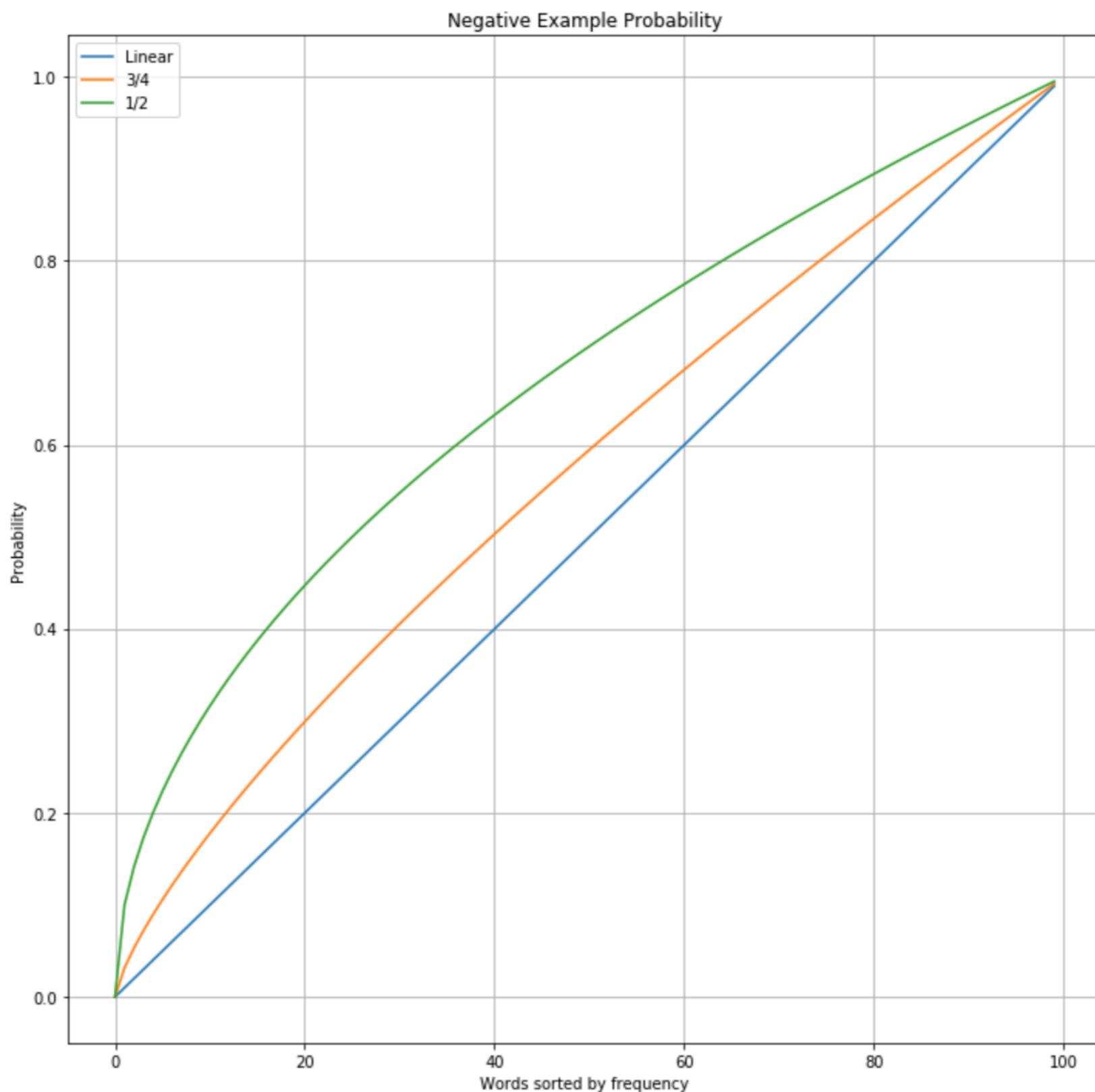
Sampling negatives

P(word)  
3/4

e.g. K = 5

Increase rare word probability

# Word2Vec



# Word2Vec

**Negative sampling**

$$J_{\text{neg-sample}}(\mathbf{o}, \mathbf{v}_c, \mathbf{U}) = -\log(\sigma(\mathbf{u}_o^\top \mathbf{v}_c)) - \sum_{k=1}^K \log(\sigma(-\mathbf{u}_k^\top \mathbf{v}_c))$$

Sampling negatives

$$\text{P(word)}^{3/4}$$

Subsampling frequent words

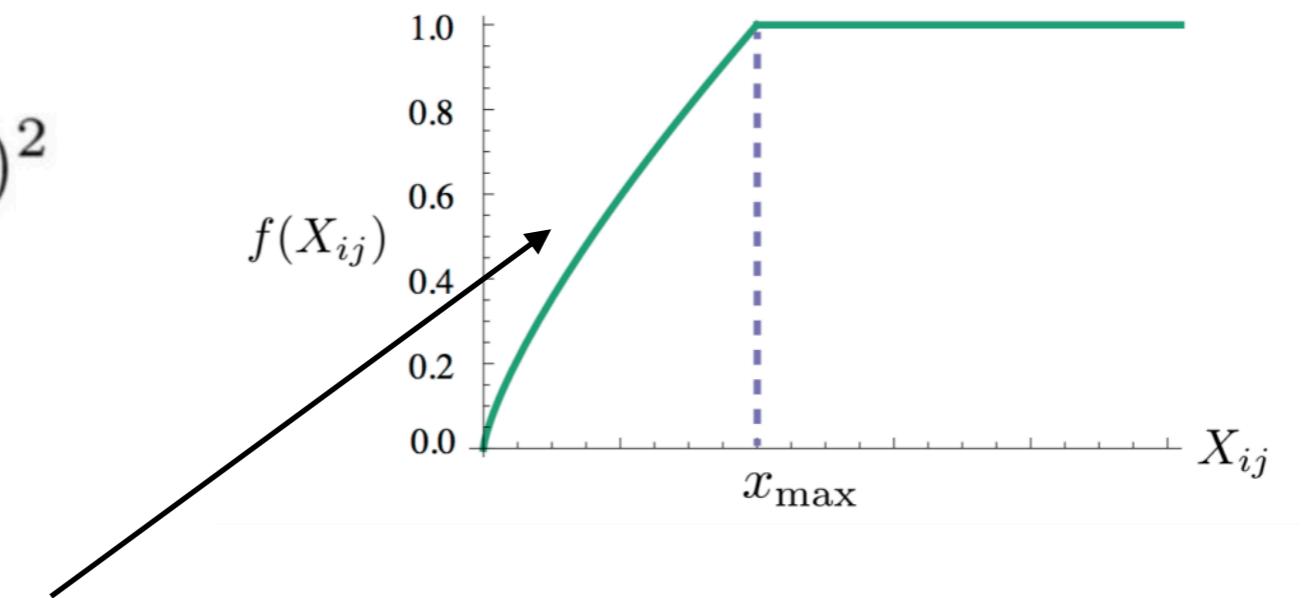
$$P(w_i) = \frac{10^{-3}}{p_i} \left( \sqrt{10^3 p_i} + 1 \right)$$

**Removing pairs**

# GloVe

$$J(\theta) = \frac{1}{2} \sum_{i,j=1}^W f(P_{ij})(u_i^T v_j - \log P_{ij})^2$$

**Less influence of rare words**

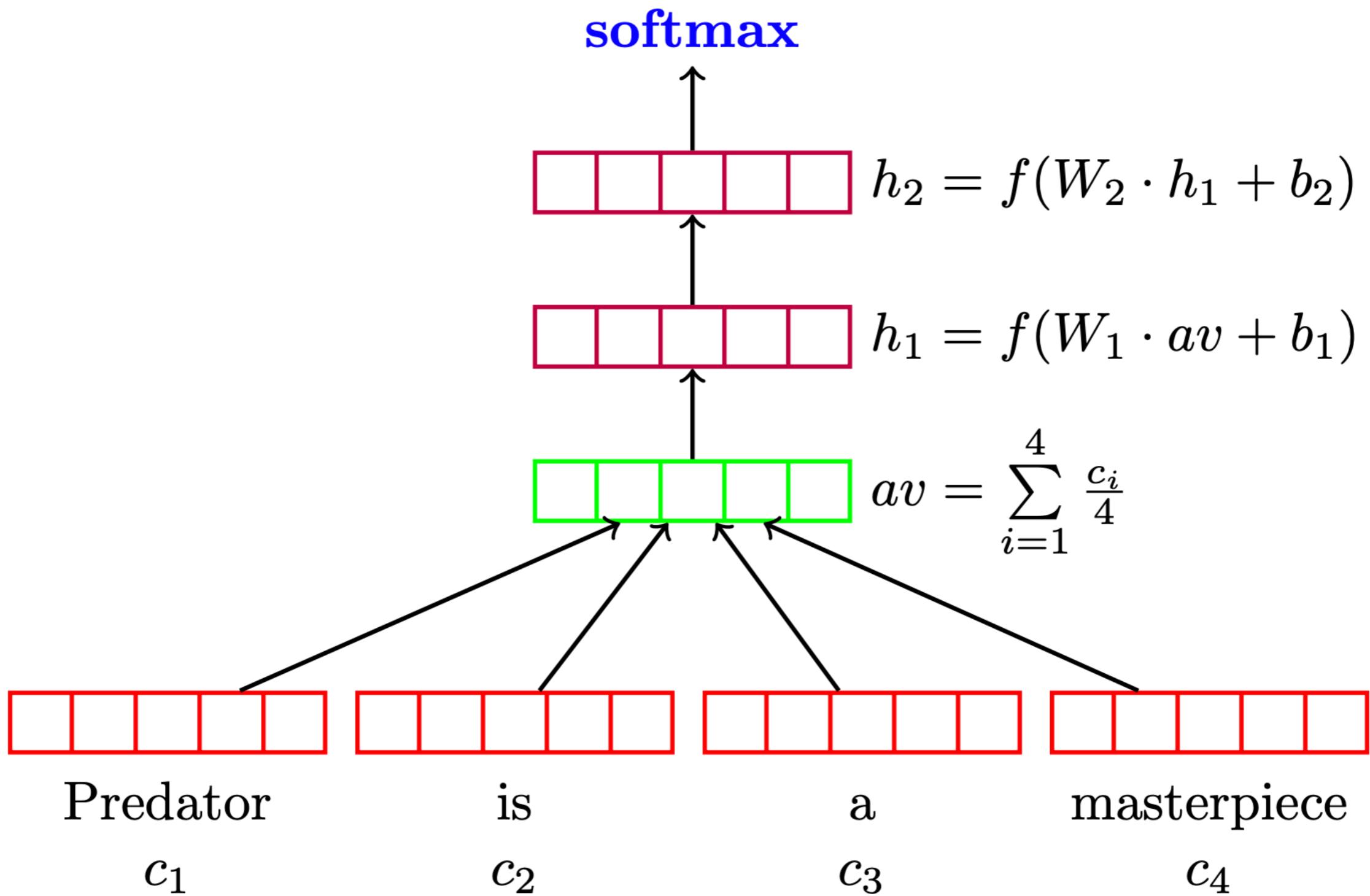


# GloVe

nearest neighbors of frog	Litoria	Leptodactylidae	Rana	Eleutherodactylus
Pictures				

# How to choose embeddings?

# DAN



# DAN

