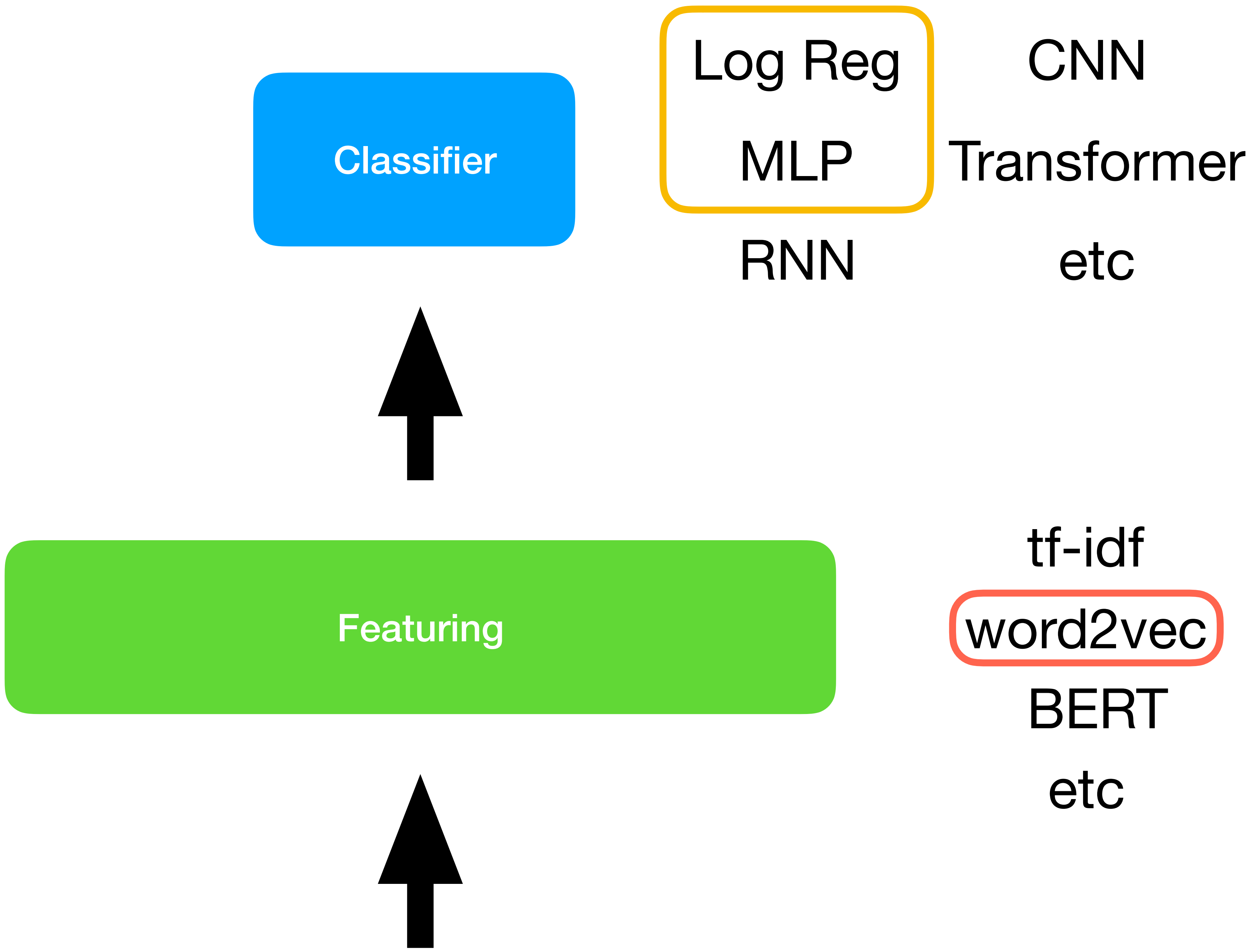
Word Embeddings

Materials MS DL Course of Boris Zubarev @bobazooba

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 0 1 0 0 0 0]
hotel = [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0]

Ortogonal vectors

Dimension = len(vocabulary)

Similarity

Dot Product

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

[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

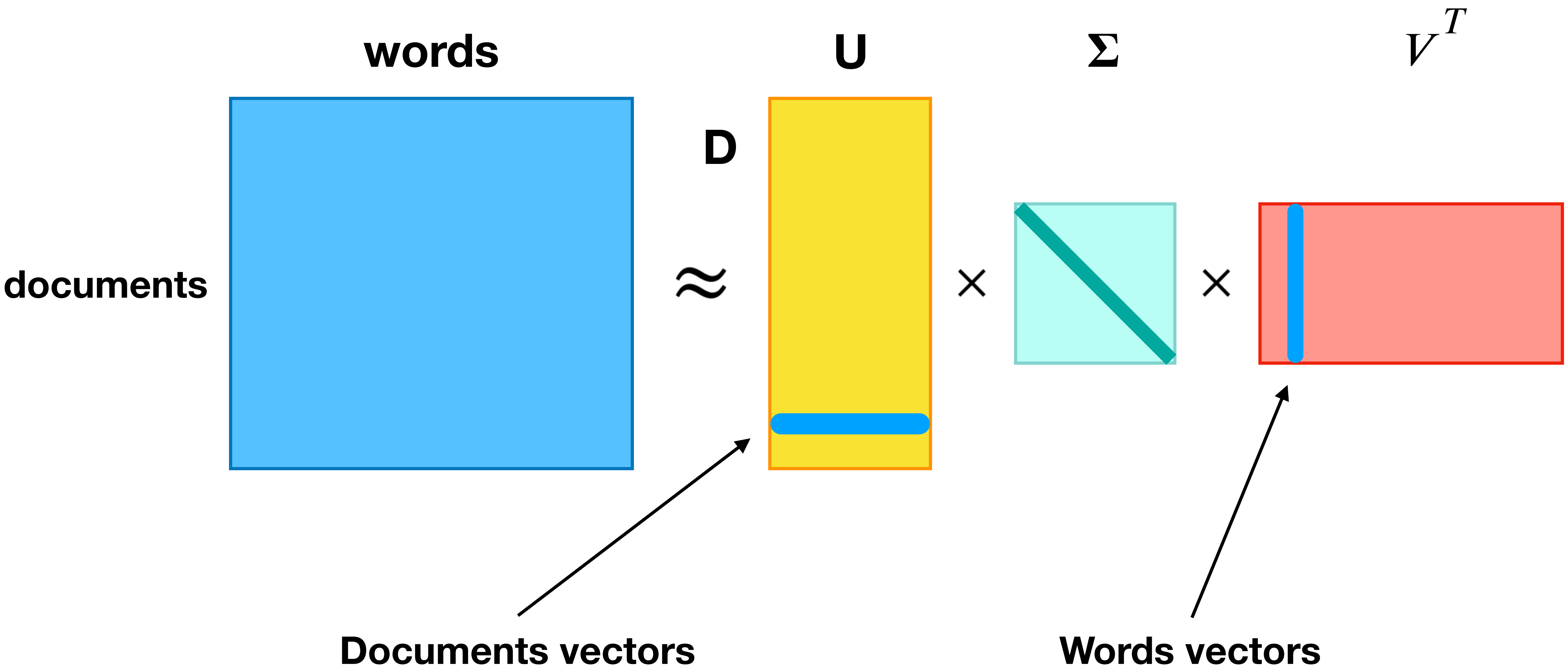
df_x = number of documents containing x

N = total number of documents

text1	0	0	0	0	0.47	0	0.23	0
text2	0	0.68	0	0	0.32	0	0	0
text3	0.11	0	0.19	0	0	0	0	0

Co-occurrence Matrix

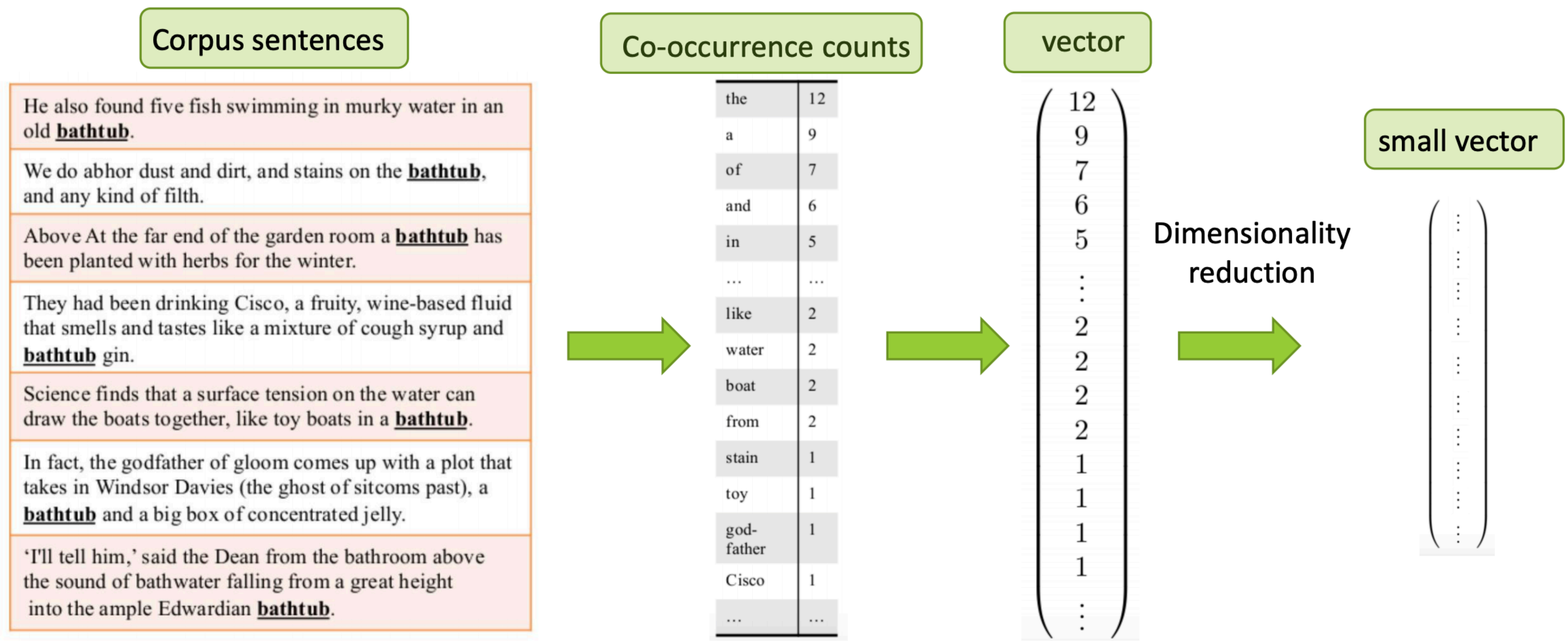
$$X \approx \hat{X} = U \, \Sigma \, V^T$$



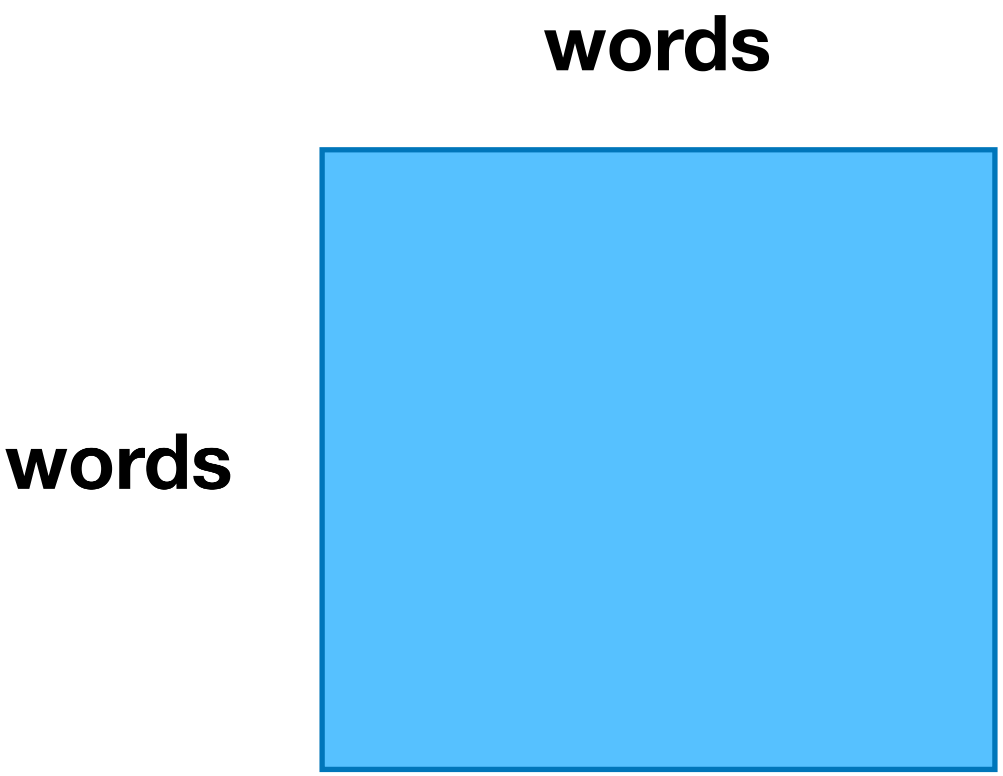
Computational expensive

Co-occurrence Vectors

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

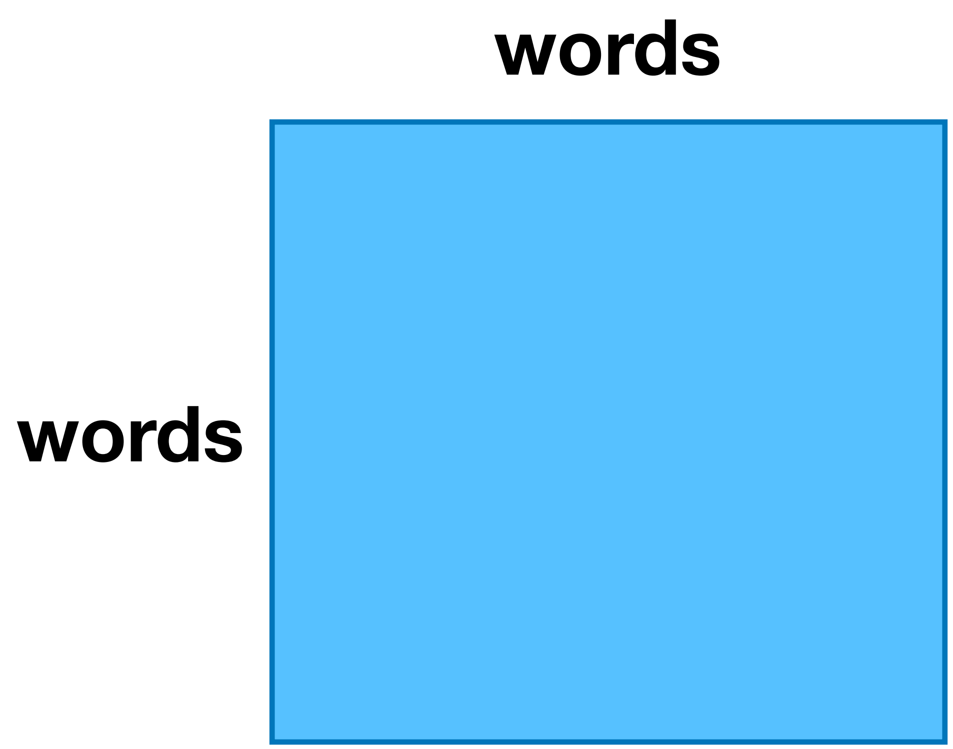


Co-occurrence Matrix



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

Co-occurrence Matrix

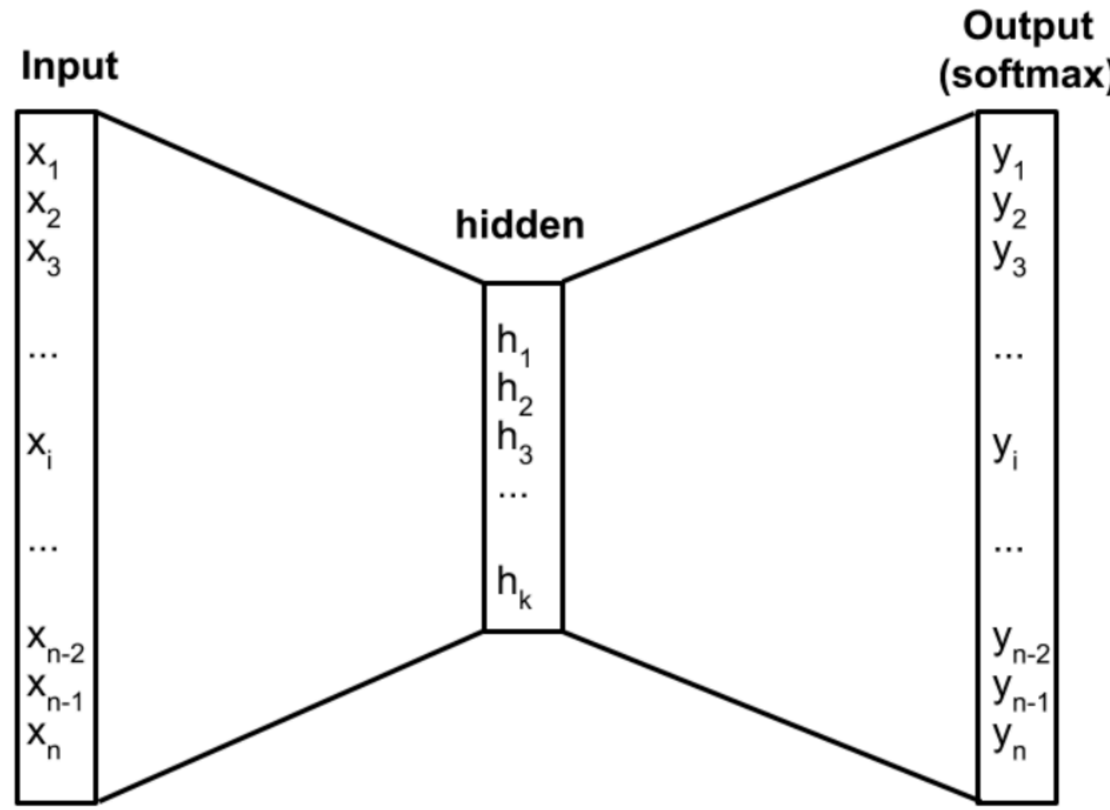


$$\mathrm{pmi}(x;y) \equiv \log\frac{p(x,y)}{p(x)p(y)} = \log\frac{p(x|y)}{p(x)} = \log\frac{p(y|x)}{p(y)}$$

ppmi = max(pmi, 0)

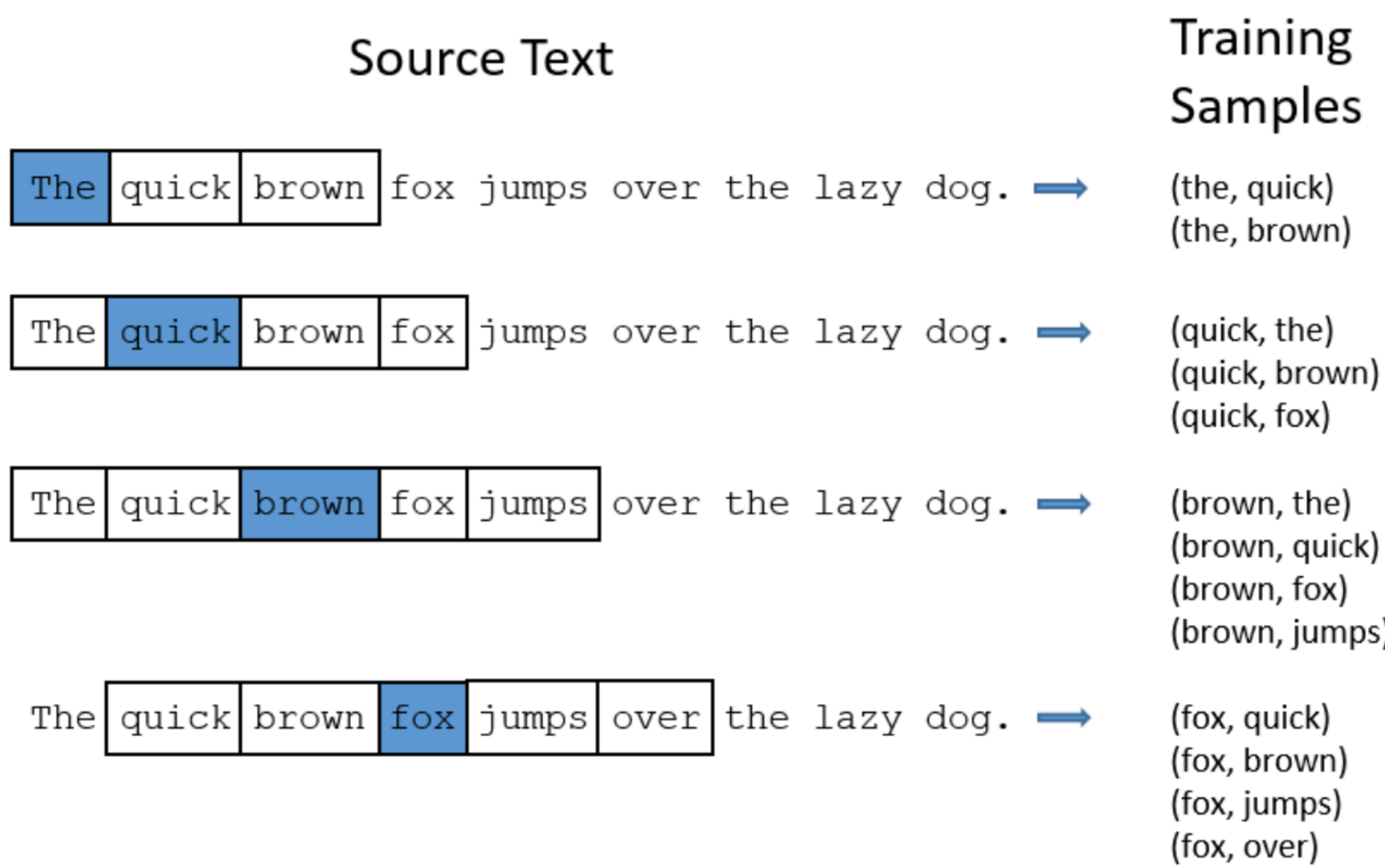
Word2Vec

Word2Vec



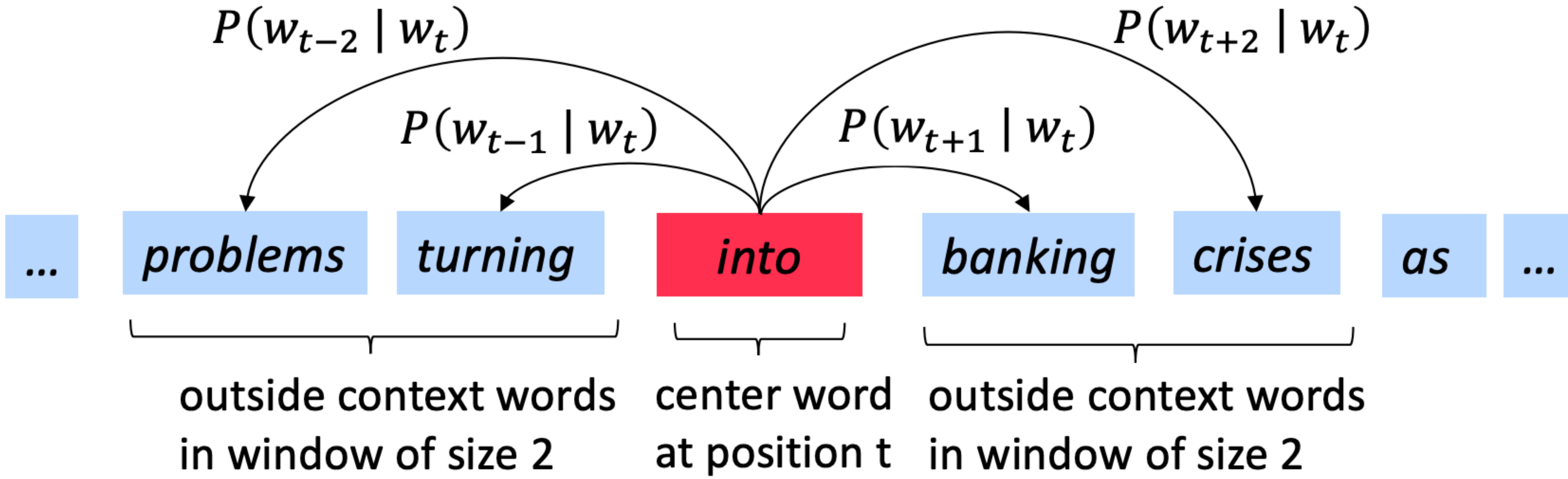
$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0} \log p(w_{t+j}|w_t)$$

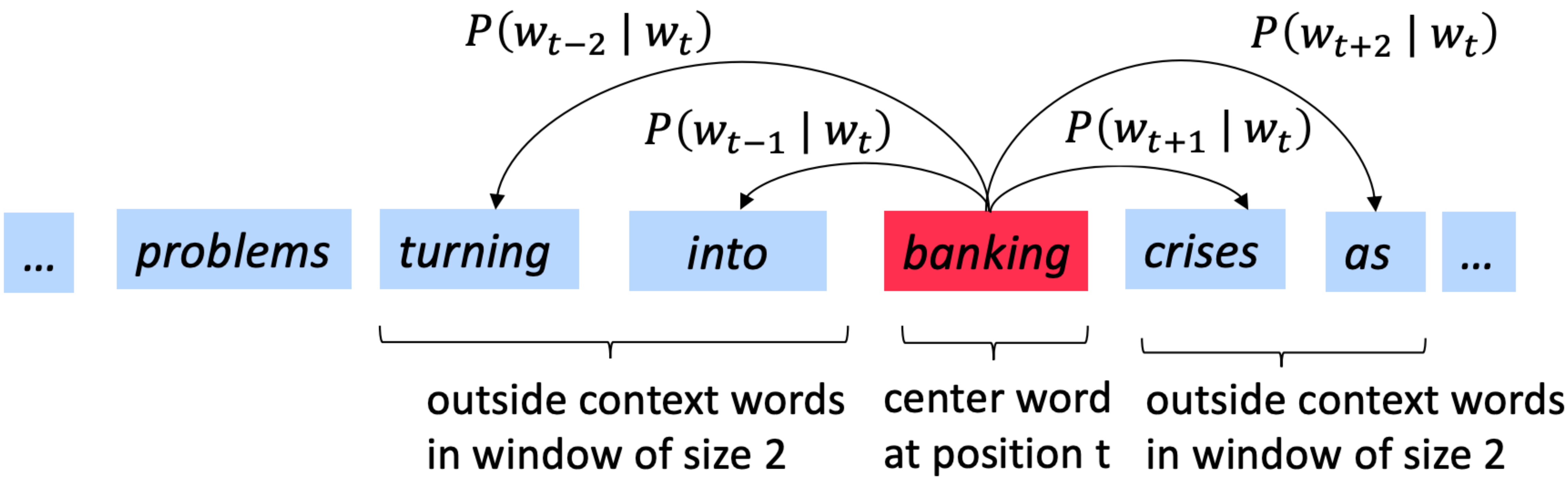
Word2Vec



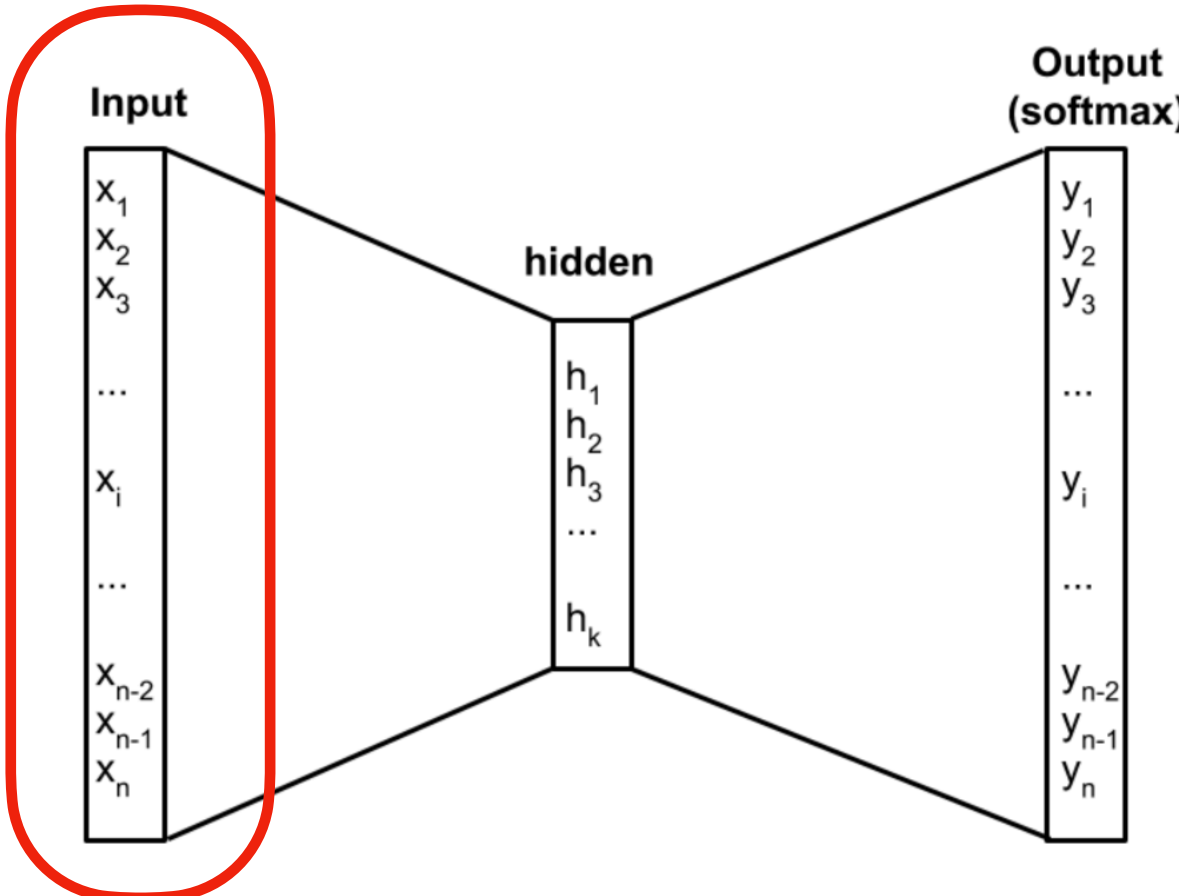
Source Text
Training Samples
The quick brown fox jumps over the lazy dog.
(the, quick)
(the, brown)
The quick brown fox jumps over the lazy dog.
(quick, the)
(quick, brown)
(quick, fox)
The quick brown fox jumps over the lazy dog.
(brown, the)
(brown, quick)
(brown, fox)
(brown, jumps)
The quick brown fox jumps over the lazy dog.
(fox, quick)
(fox, brown)
(fox, jumps)
(fox, over)

Word2Vec


$P(w_{t-2} \mid w_t)$
$P(w_{t-1} \mid w_t)$
$P(w_{t+1} \mid w_t)$
$P(w_{t+2} \mid w_t)$
...
problems
turning
into
banking
crises
as
...
outside context words in window of size 2
center word at position t
outside context words in window of size 2

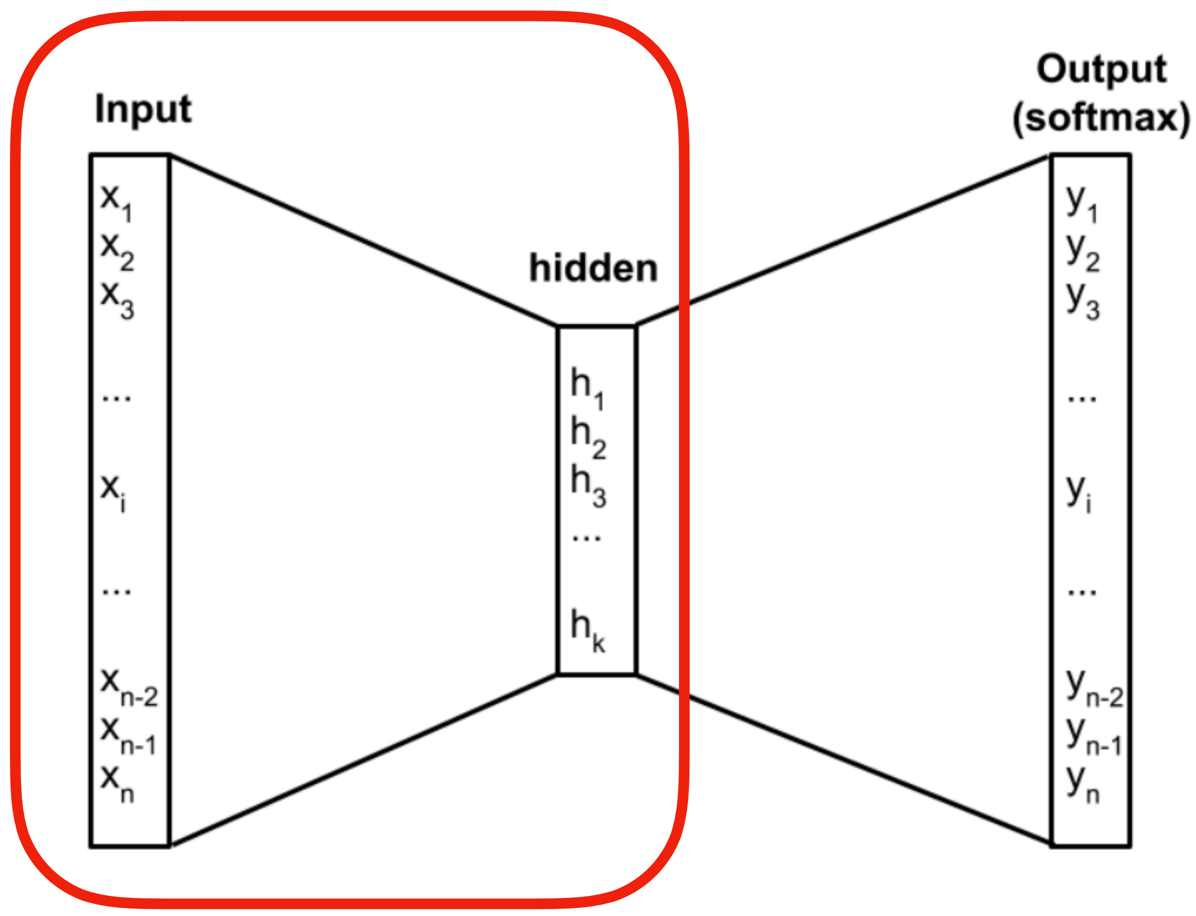

$P(w_{t-2} \mid w_t)$
$P(w_{t-1} \mid w_t)$
$P(w_{t+1} \mid w_t)$
$P(w_{t+2} \mid w_t)$
...
problems
turning
into
banking
crises
as
...
outside context words in window of size 2
center word at position t
outside context words in window of size 2

Word2Vec



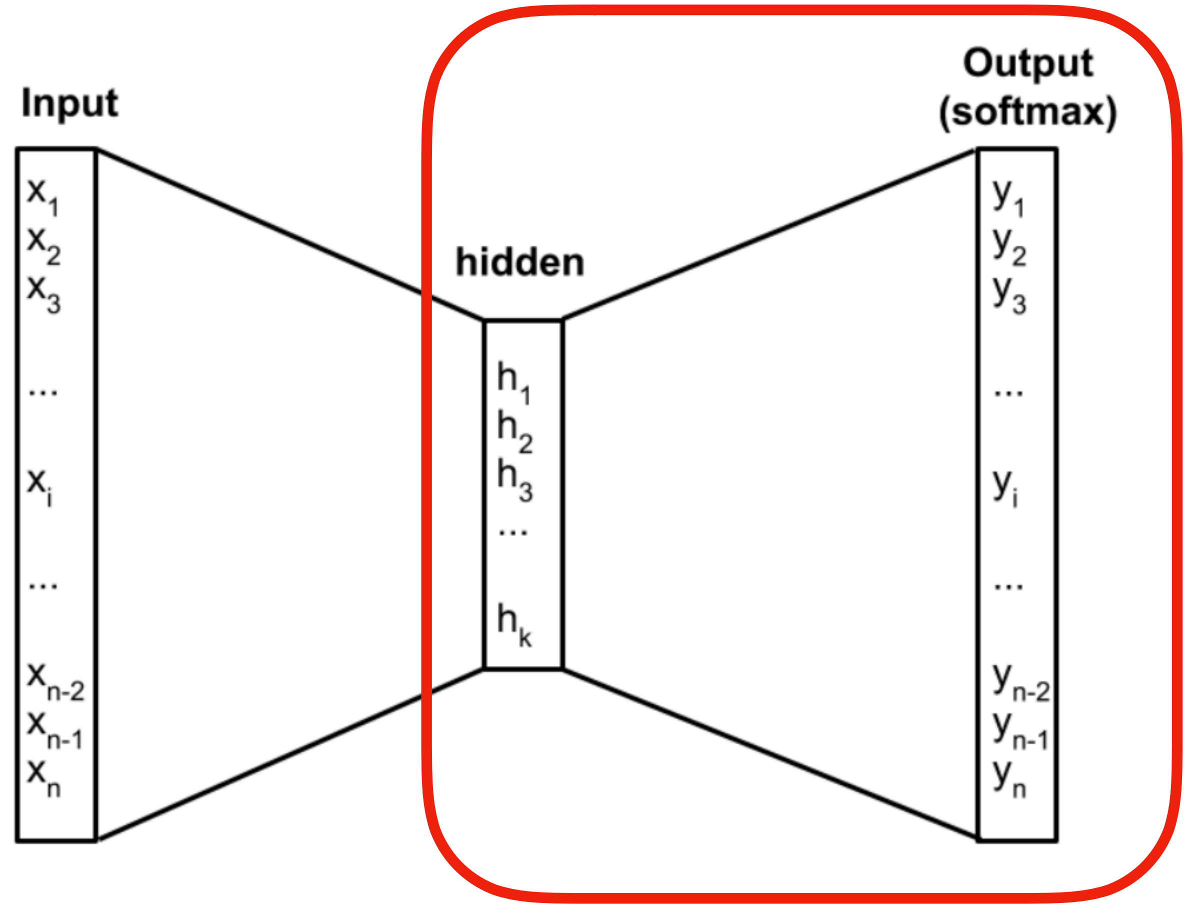
$$\theta = \begin{bmatrix} v_{aardvark} \\ v_a \\ \vdots \\ v_{zebra} \\ u_{aardvark} \\ u_a \\ \vdots \\ u_{zebra} \end{bmatrix} \in \mathbb{R}^{2dV}$$

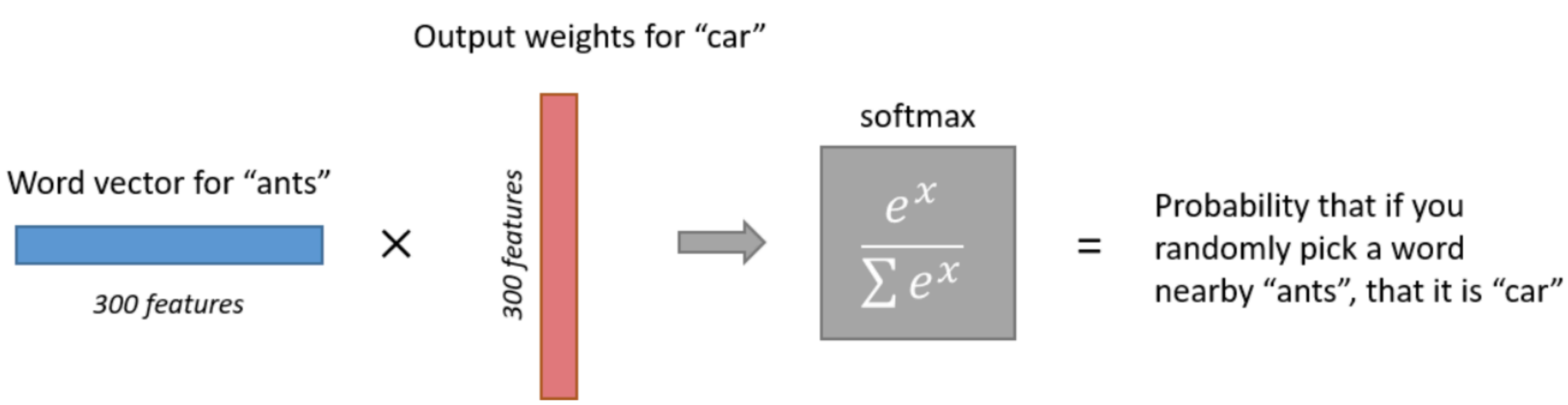
Word2Vec



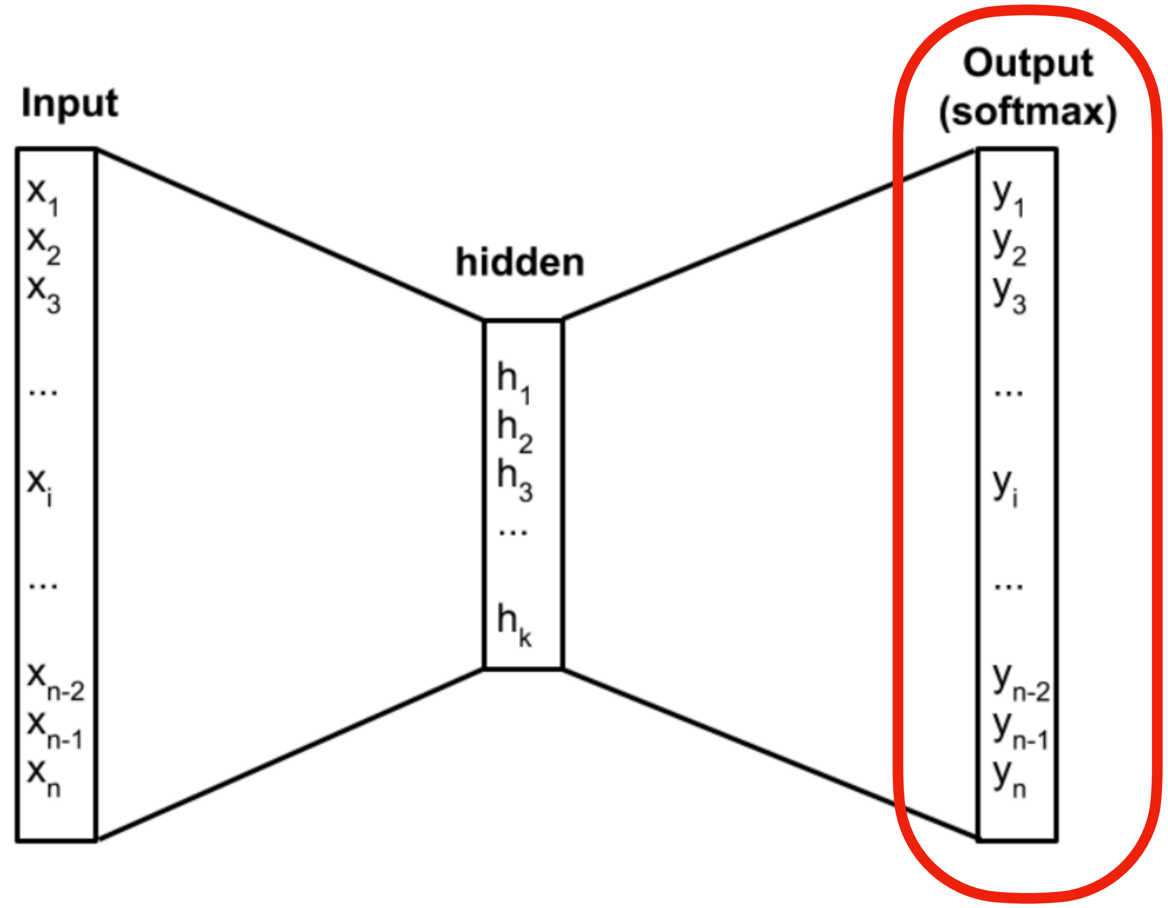
$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} = \begin{bmatrix} 10 & 12 & 19 \end{bmatrix}$$

Word2Vec


Input
Output
(softmax)
hidden
x_1 x_2 x_3 … x_i … x_{n-2} x_{n-1} x_n
h_1 h_2 h_3 … h_k
y_1 y_2 y_3 … y_i … y_{n-2} y_{n-1} y_n

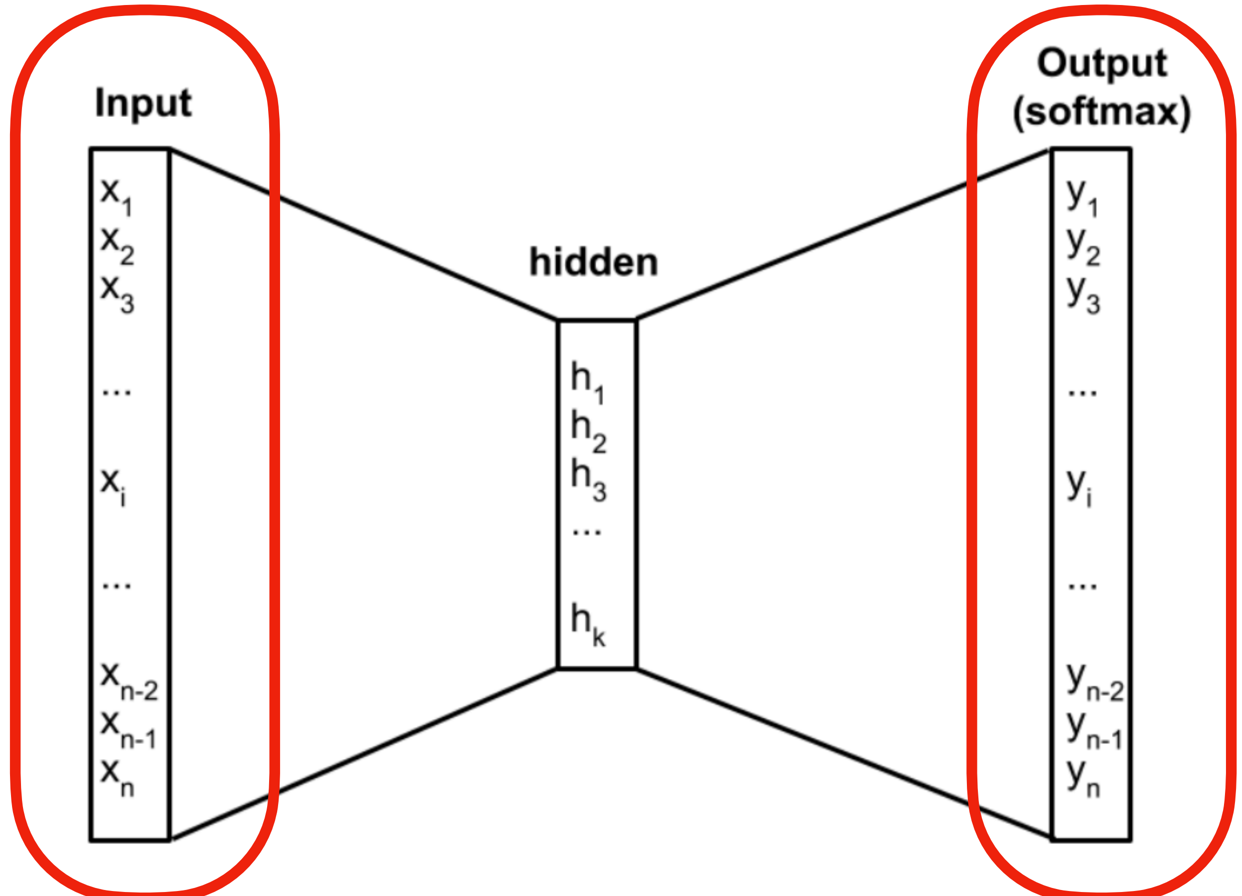

Output weights for "car"
Word vector for "ants"
300 features
×
300 features
softmax
$\frac{e^x}{\sum e^x}$
=
Probability that if you randomly pick a word nearby "ants", that it is "car"

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



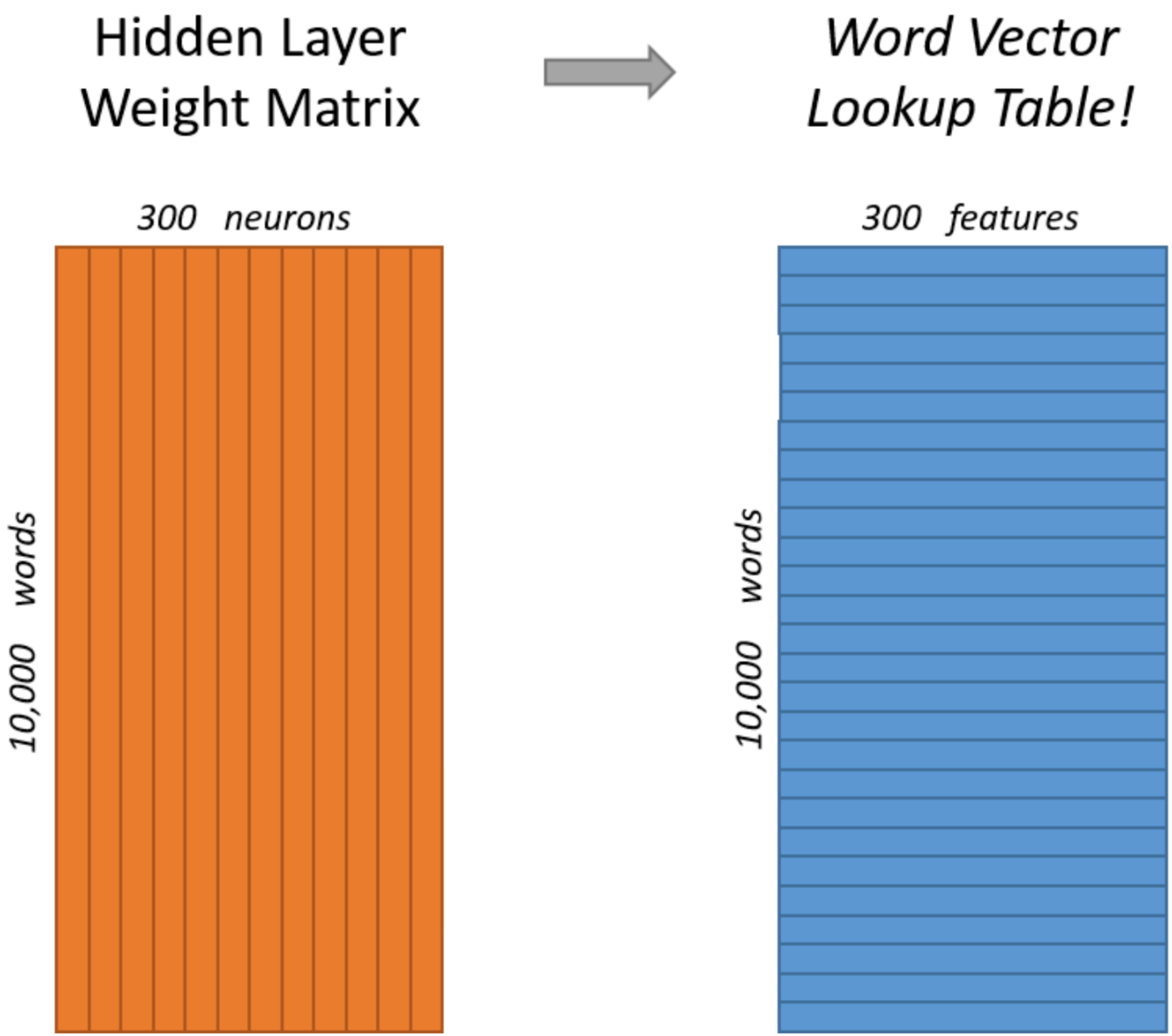
$$\theta = \begin{bmatrix} v_{aardvark} \\ v_a \\ \vdots \\ v_{zebra} \\ u_{aardvark} \\ u_a \\ \vdots \\ u_{zebra} \end{bmatrix} \in \mathbb{R}^{2dV}$$

Same words

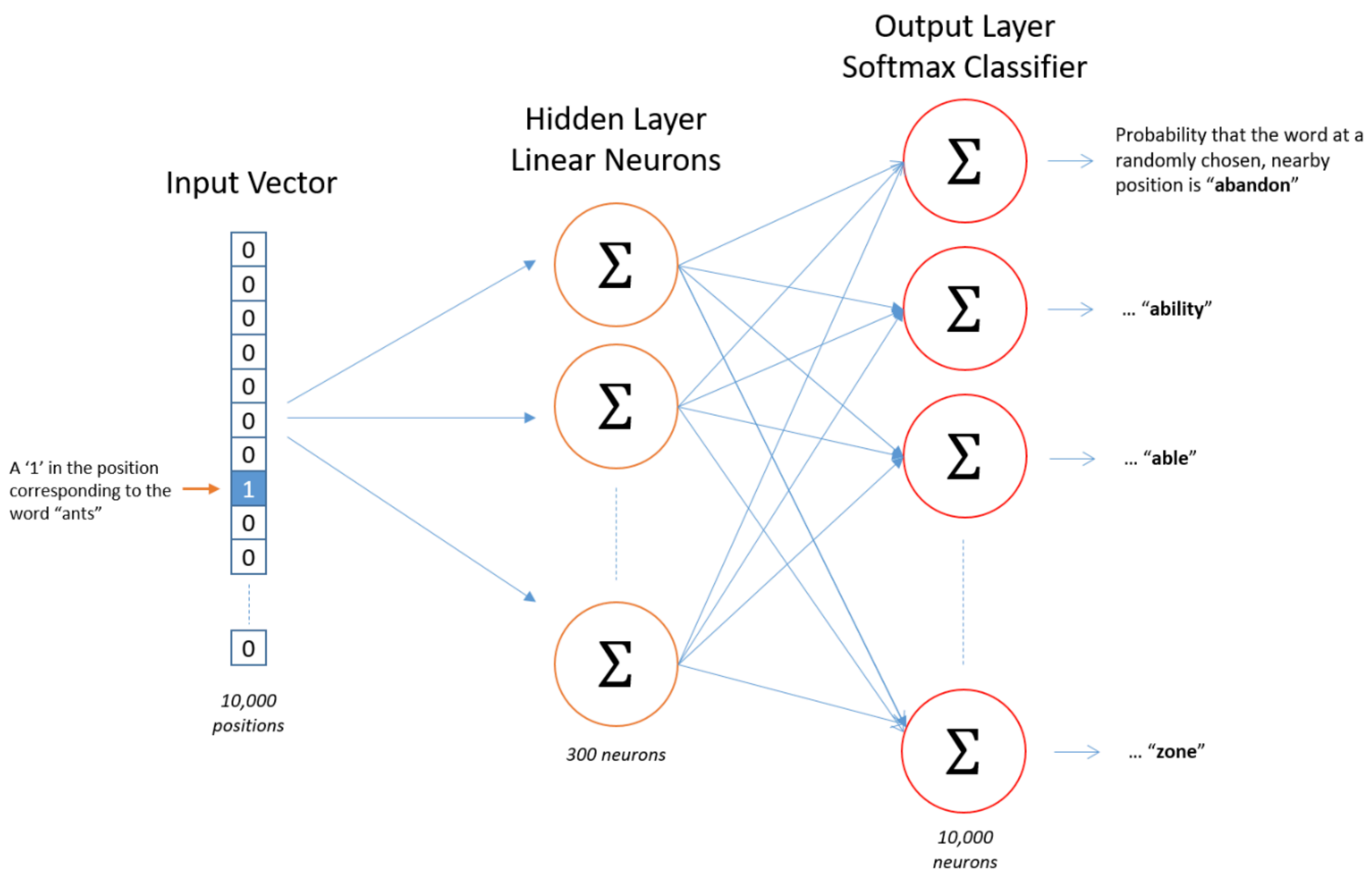
Different vectors

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

Word2Vec


Hidden Layer
Weight Matrix
Word Vector
Lookup Table!
300 neurons
10,000 words
300 features
10,000 words

Word2Vec

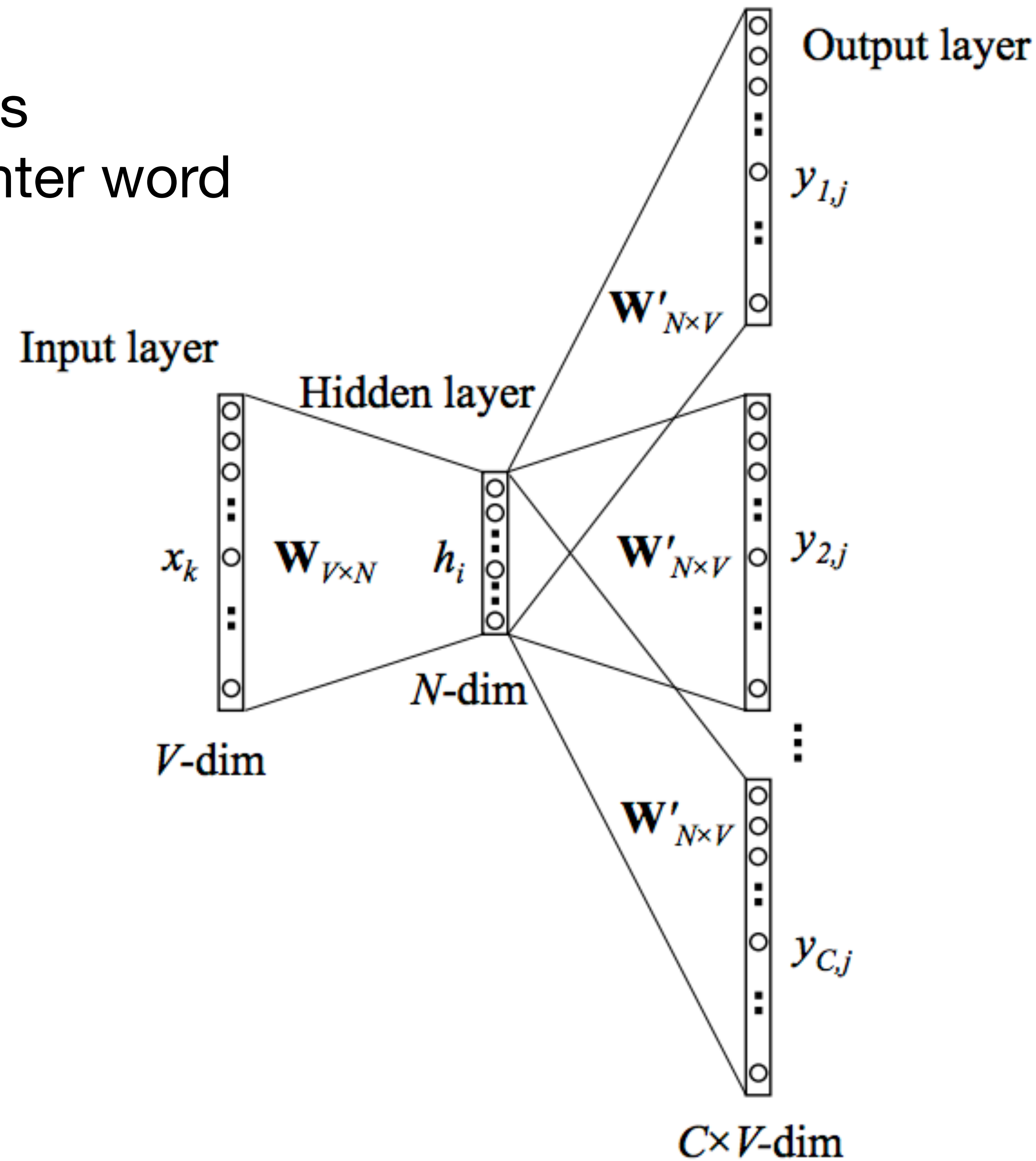

Output Layer
Softmax Classifier
Hidden Layer
Linear Neurons
Input Vector
Probability that the word at a randomly chosen, nearby position is "**abandon**"
... "**ability**"
... "**able**"
... "**zone**"
A '1' in the position corresponding to the word "ants"
0
0
0
0
0
0
0
1
0
0
0
10,000 positions
300 neurons
10,000 neurons

Word2Vec

Skipgrams

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

Better for rare words

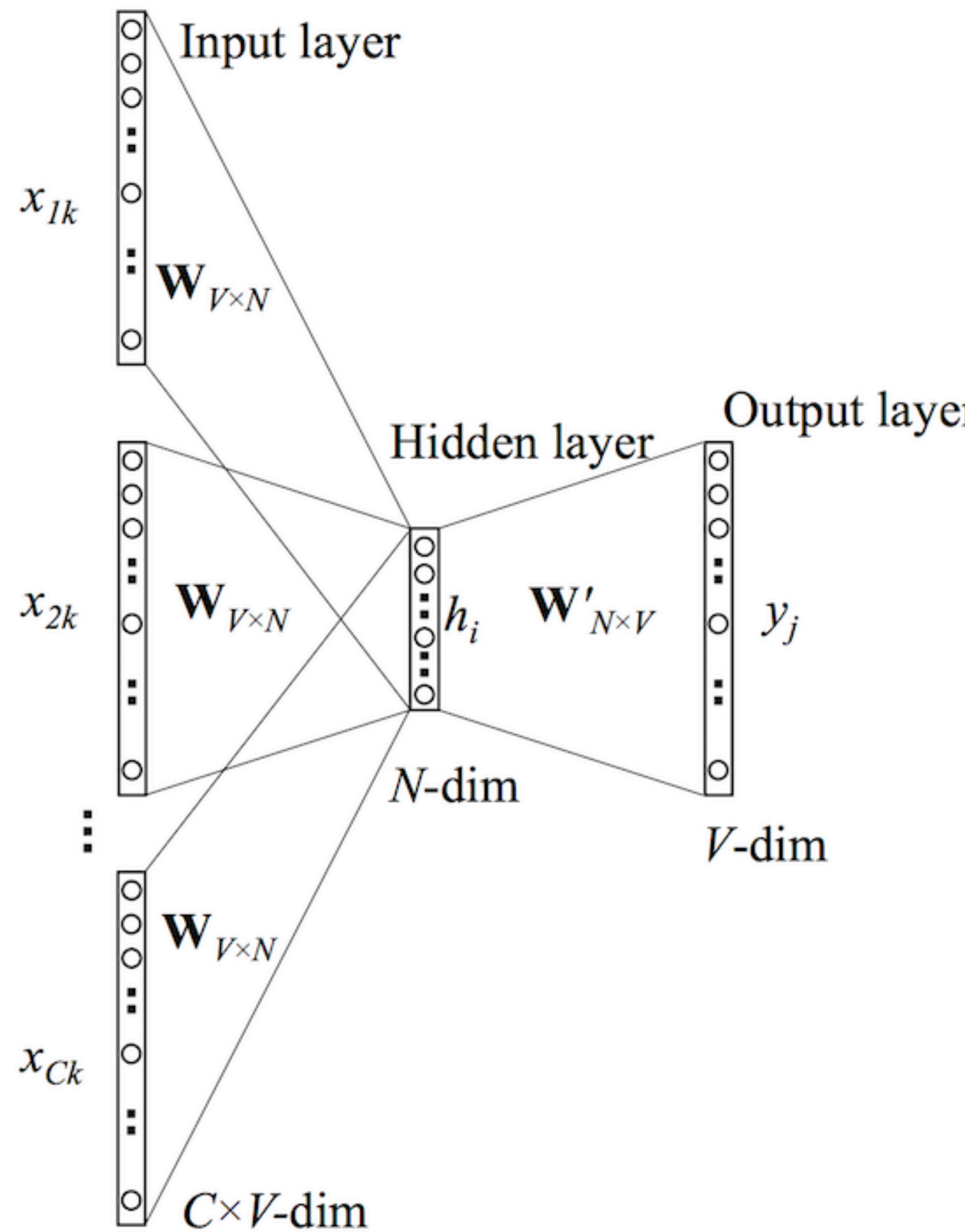


Word2Vec

CBOW

Predict center word from (bag of) context words

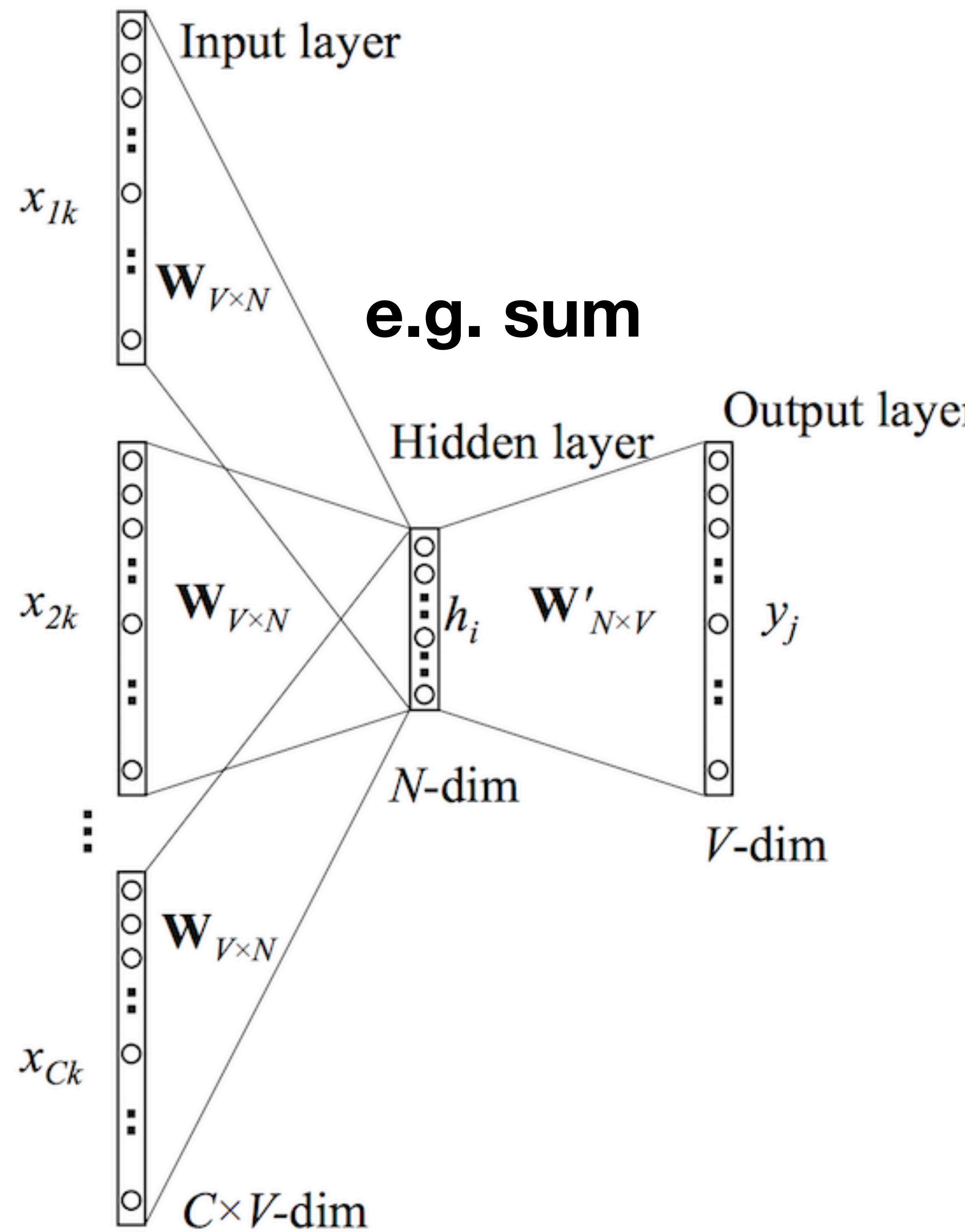
Faster



Word2Vec

CBOW

Predict center word from
(bag of) context words



Visualization

https://projector.tensorflow.org/

- BERT Embedding Projector

Visualization

Москва_PROPN ↓ Россия_PROPN

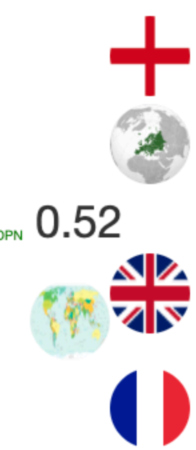
Лондон_PROPN ↓ ???

Частотность слова

- [x] Высокая
- [x] Средняя
- [ ] Низкая

НКРЯ и Wikipedia

1. англия PROPN 0.58
2. европа PROPN 0.54
3. великобритания PROPN 0.52
4. страна NOUN 0.48
5. франция PROPN 0.47



Visualization

Some vector close to queen

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

Word2Vec

Fasttext

OOV

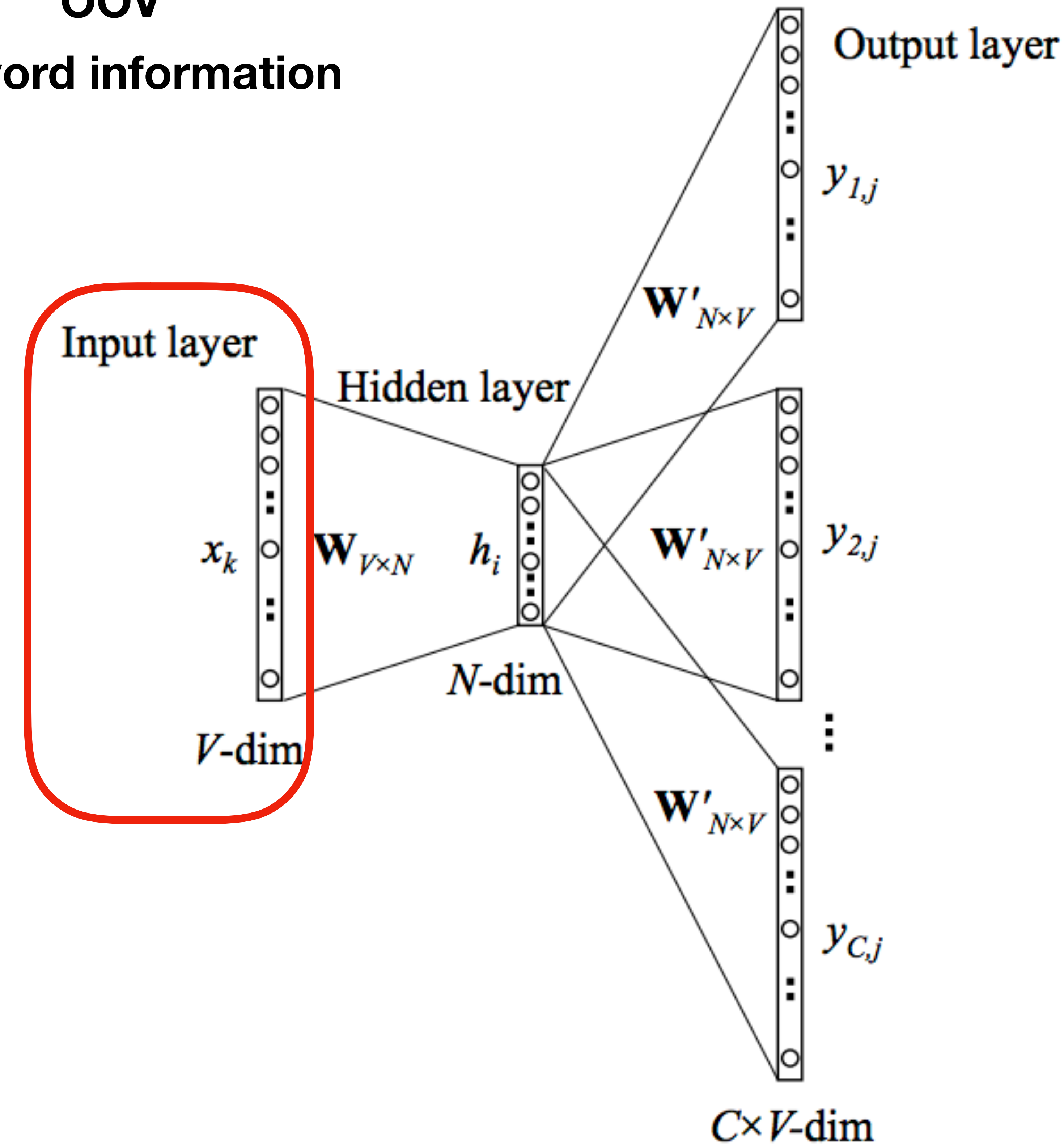
Subword information

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

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

Computational expensive

Word2Vec

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

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

Word2Vec

Negative sampling

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

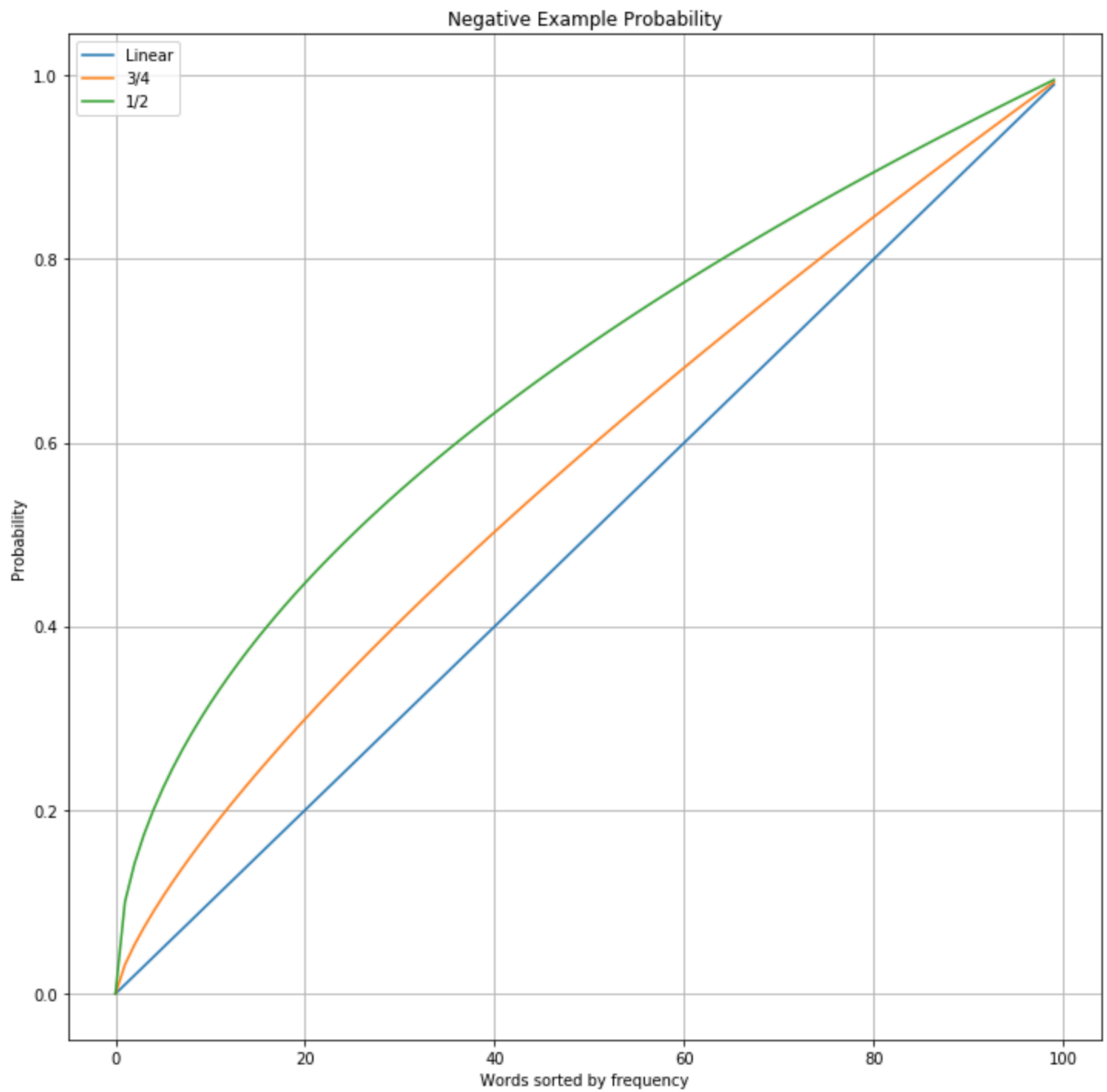
Sampling negatives

e.g. K = 5

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

Increase rare word probability

Word2Vec



Negative Example Probability
Linear
3/4
1/2
1.0
0.8
0.6
0.4
0.2
0.0
Probability
0
20
40
60
80
100
Words sorted by frequency

Word2Vec

Negative sampling

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

Sampling negatives

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

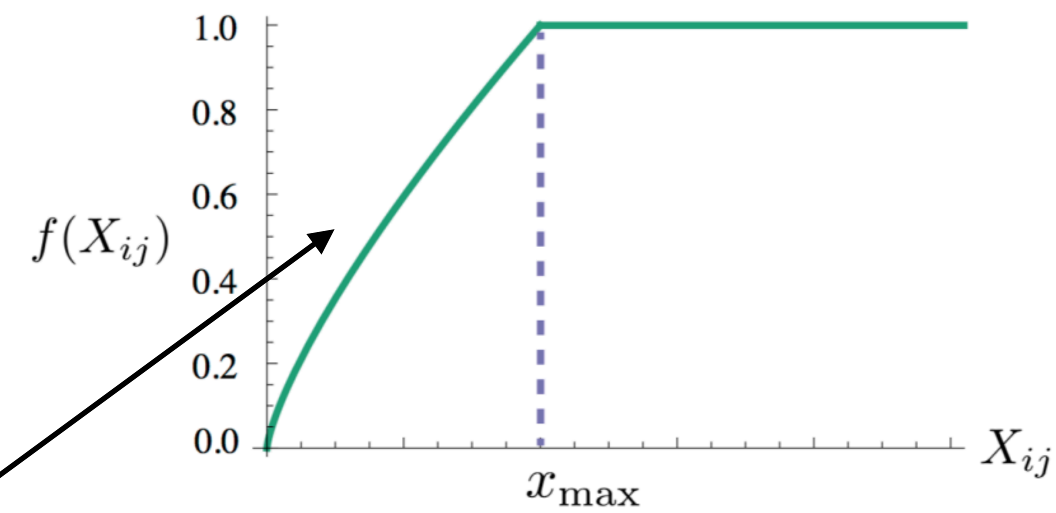
Subsampling frequent words

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

Removing pairs

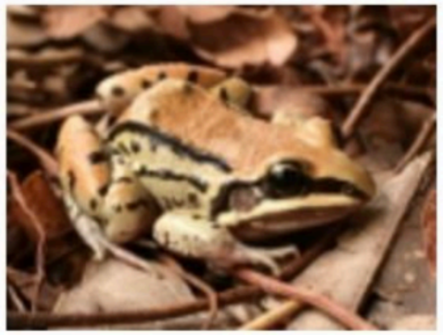
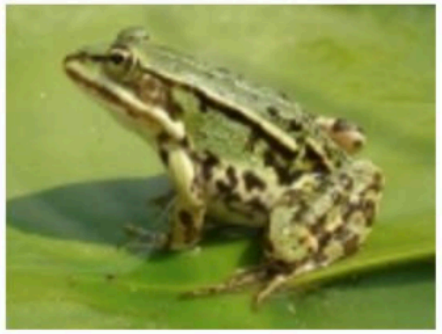
GloVe

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



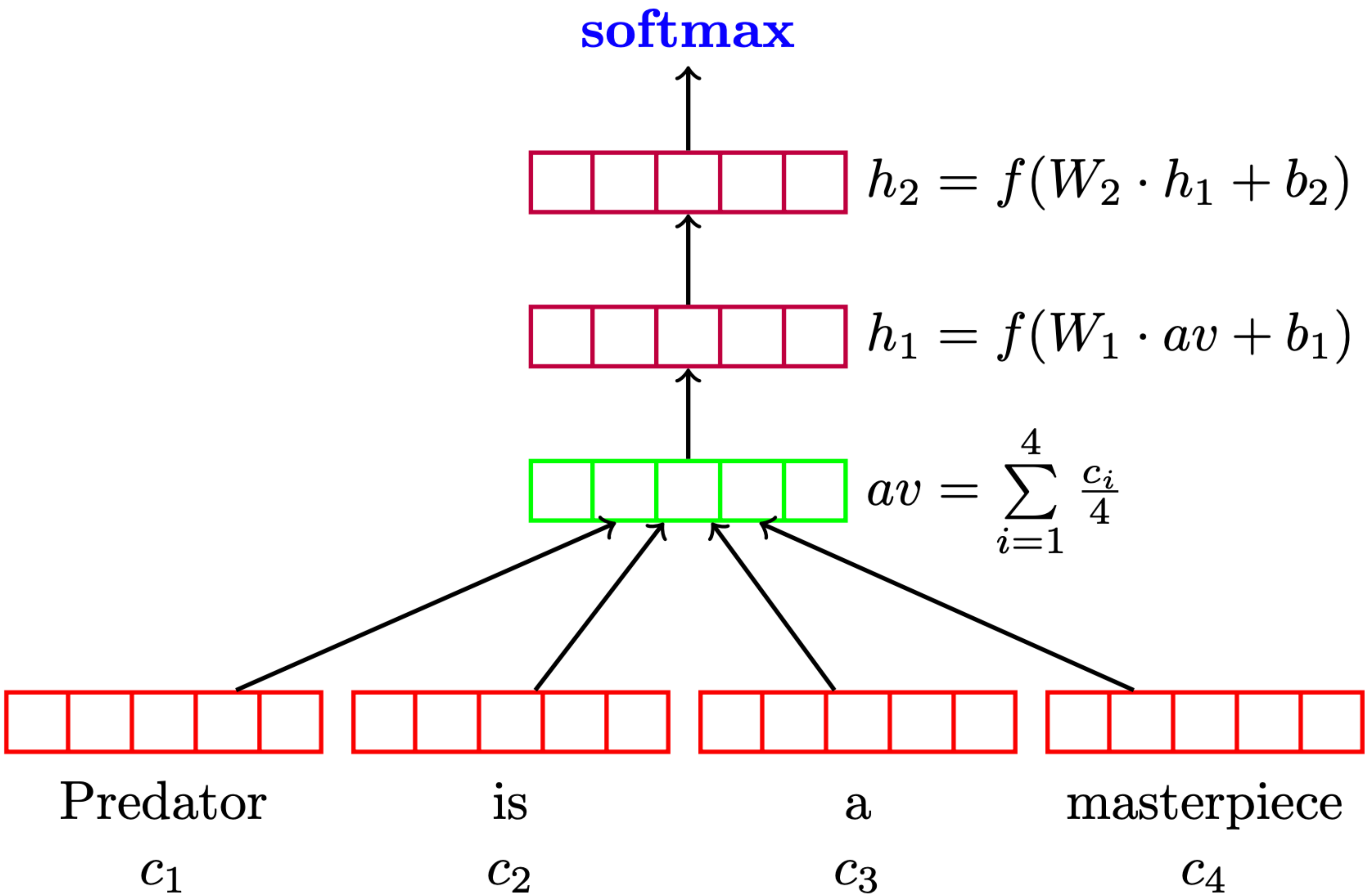
Less influence of rare words

GloVe

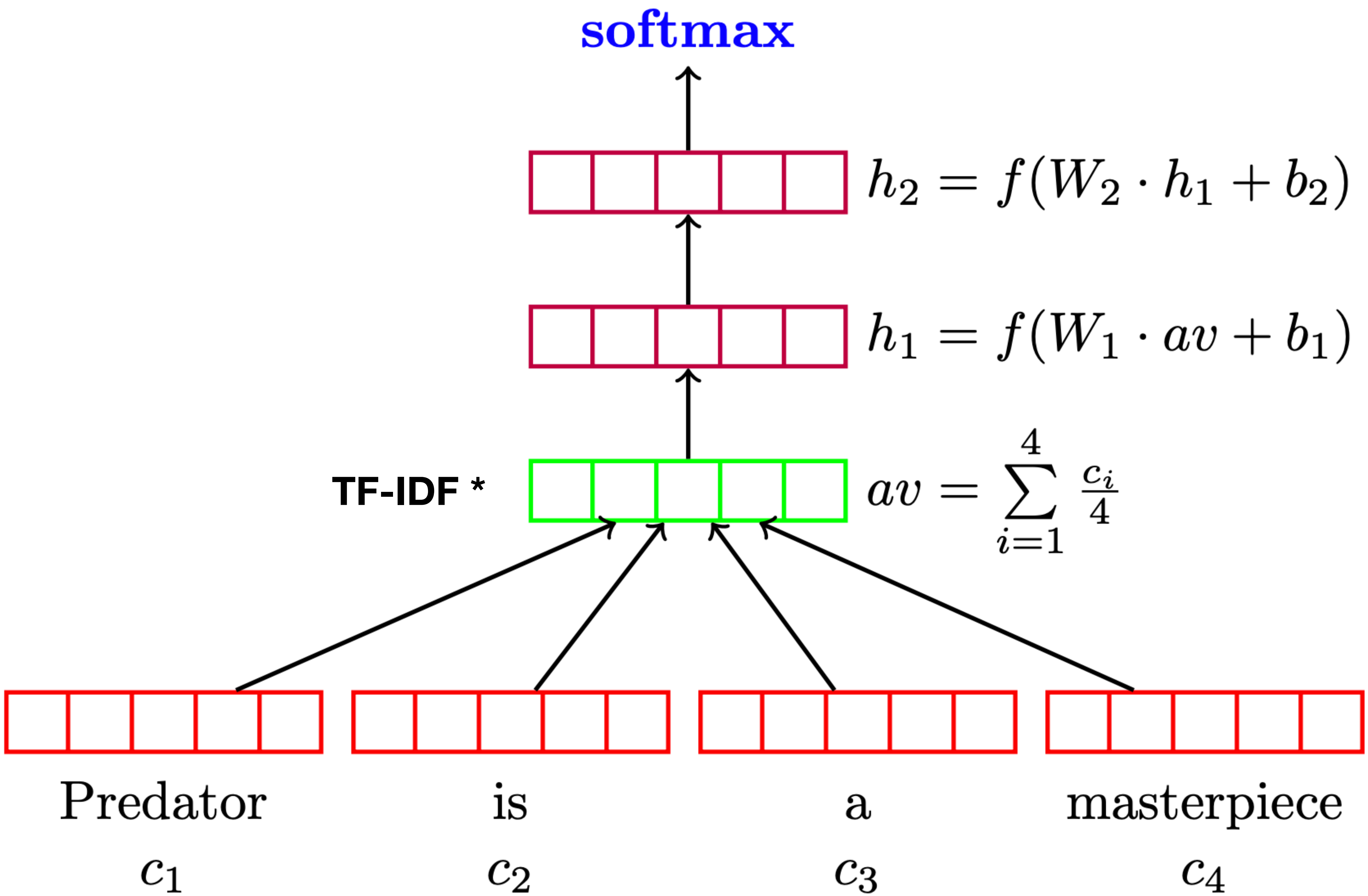
nearest neighbors of *frog*	Litoria	Leptodactylidae	Rana	Eleutherodactylus
Pictures				

How to choose embeddings?

DAN


softmax
$h_2 = f(W_2 \cdot h_1 + b_2)$
$h_1 = f(W_1 \cdot av + b_1)$
$av = \sum_{i=1}^{4} \frac{c_i}{4}$
Predator
is
a
masterpiece
c_1
c_2
c_3
c_4

DAN



softmax
$h_2 = f(W_2 \cdot h_1 + b_2)$
$h_1 = f(W_1 \cdot av + b_1)$
TF-IDF *
$av = \sum_{i=1}^{4} \frac{c_i}{4}$
Predator
is
a
masterpiece
c_1
c_2
c_3
c_4