Natural Language Processing
Language Models

Sudeshna Sarkar

24 July 2019

Neural language model

$$P(w_t|w_{t-n},\dots,w_{t-1}) = \cancel{\frac{C(w_{t-n},\dots,w_t)}{C(w_{t-n},\dots,w_{t-1})}}$$

$$= f_\theta(w_{t-n},\dots,w_{t-1})$$

- Parametric estimator
- We need numerical representation of words

"A word is known by the company it keeps"

Word Representation

- Continuous Representation: based on context
- Distributional hypothesis

You can get a lot of value by representing a word by means of its neighbors

"You shall know a word by the company it keeps"

(J. R. Firth 1957: 11)

One of the most successful ideas of modern NLP

government debt problems turning into banking crises as has happened in

saying that Europe needs unified banking regulation to replace the hodgepodge

↖ These words will represent *banking* ↗

Representation

- We need effective representation of :
  - Words, Sentences, Text

1: Use existing thesauri or ontologies like WordNet

Drawbacks:

- Manual
- Not context specific

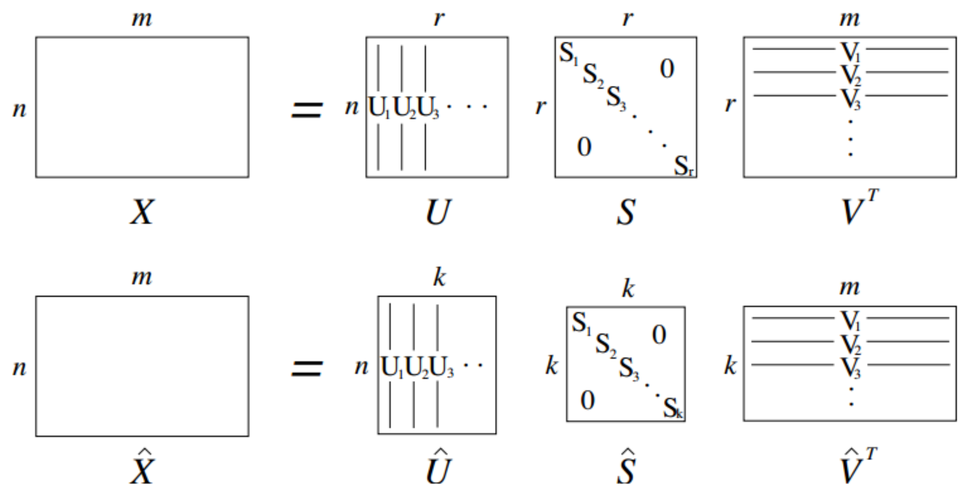
2: Use co-occurrences for word similarity. Drawbacks:

- Quadratic space needed
- Relative position and order of words not considered

co-occurrence

1. Use co-occurrences for word similarity.
2. Singular Value Decomposition (SVD) on co-occurrence matrix
   - $\hat{X}$ is the best rank k approximation to X , in terms of least squares

- $m = n =$
  size of vocabulary



- $\hat{S}$ is the same matrix as S except that it contains only the top largest singular values

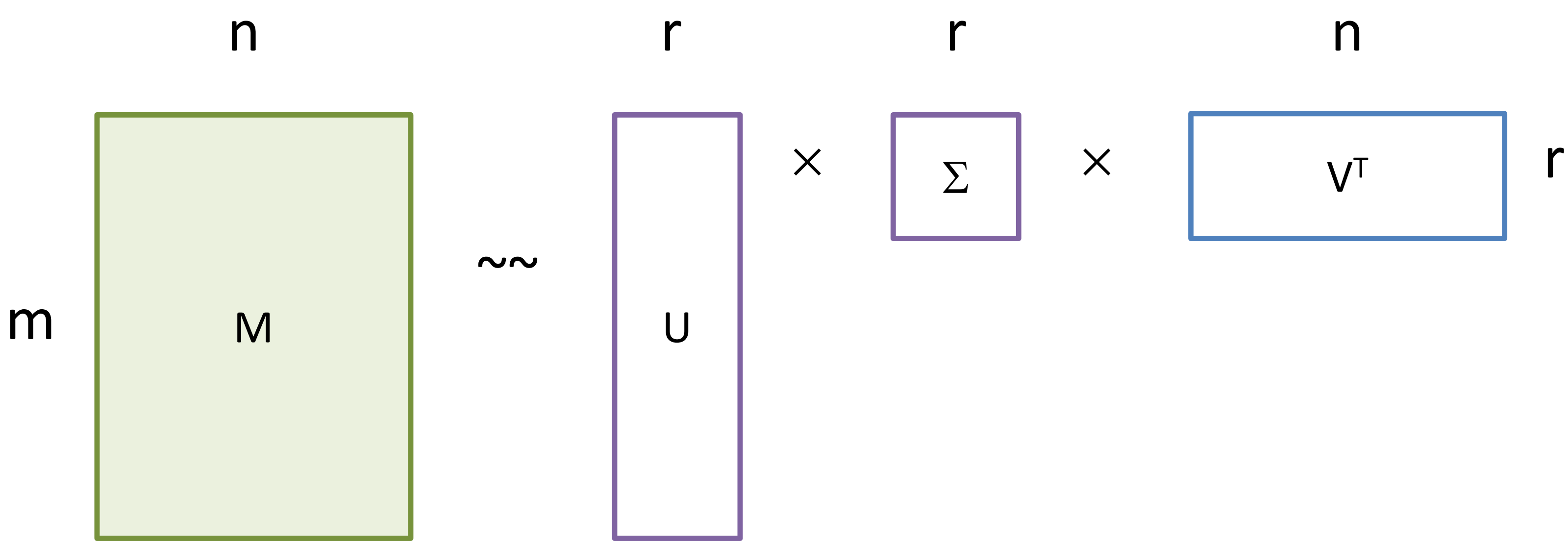
SVD

- A decomposition of any matrix into a product of three matrices.
- From this decomposition, you can choose any number r of intermediate concepts (latent factors) in a way that minimizes the RMSE error given that value of r.

- The rank of a matrix is the maximum number of rows (or equivalently columns) that are linearly independent.
- If a matrix has rank r, then it can be decomposed exactly into matrices whose shared dimension is r.

- Vectors are orthogonal if their dot product is 0.
- An orthonormal basis is a set of unit vectors any two of which are orthogonal.

Form of SVD



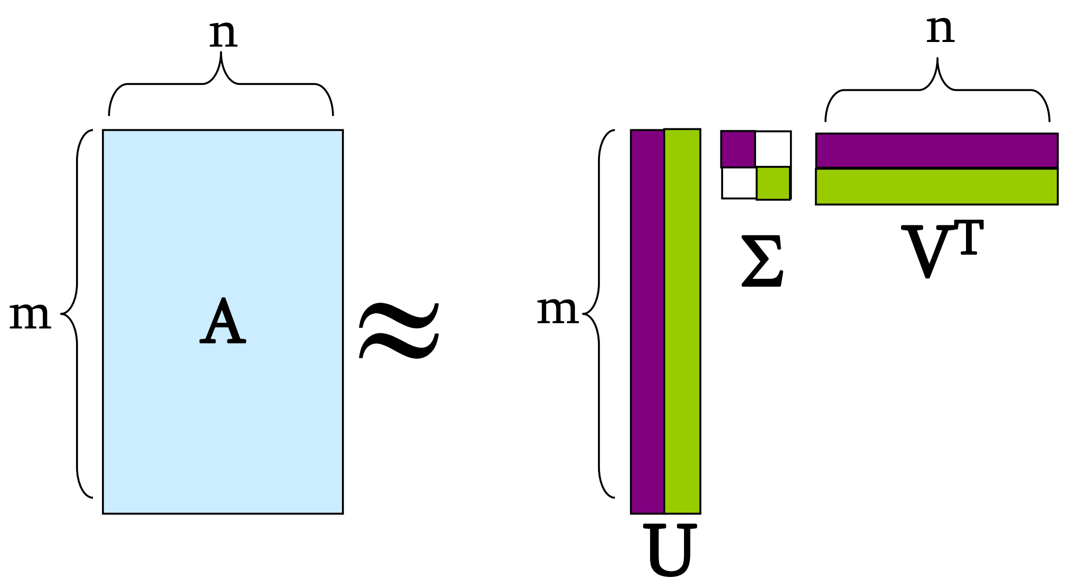
Special conditions:

U and V are column-orthonormal
(so V^T has orthonormal rows)
Σ is a diagonal matrix
The values of Σ along the diagonal are called the *singular values*.

- It is possible to decompose M exactly, if r is the rank of M.
- But we may make r much smaller, by setting to 0 the smallest singular values.
  – making the corresponding columns of U and V useless

Linkage Among Components of U, V, Σ

$$\mathbf{A} \approx \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_i \sigma_i \mathbf{u}_i \circ \mathbf{v}_i^{\mathsf{T}}$$



word2vec approach to represent the meaning of word

- Represent each word with a low-dimensional vector
- Word similarity = vector similarity
- Key idea: Predict surrounding words of every word
- Faster and can easily incorporate a new sentence/document or add a word to the vocabulary

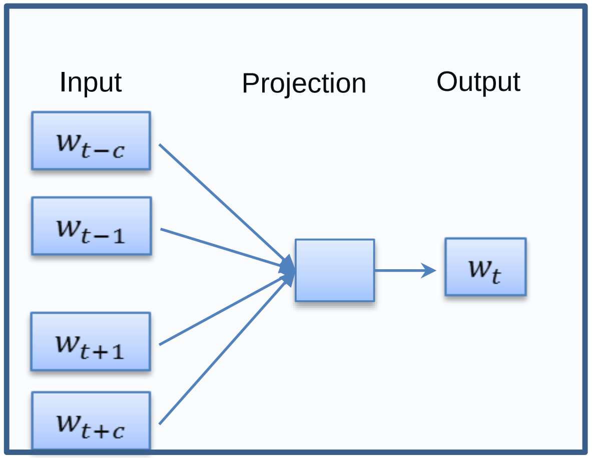
Word2vec

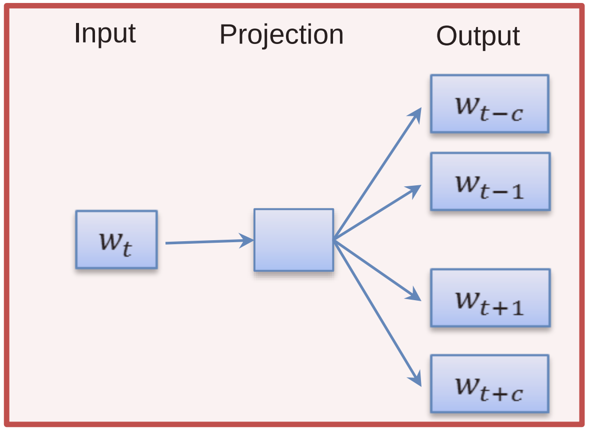
- Representation of words

``Similar words have similar contexts”

1. CBOW: P(Word|Context)

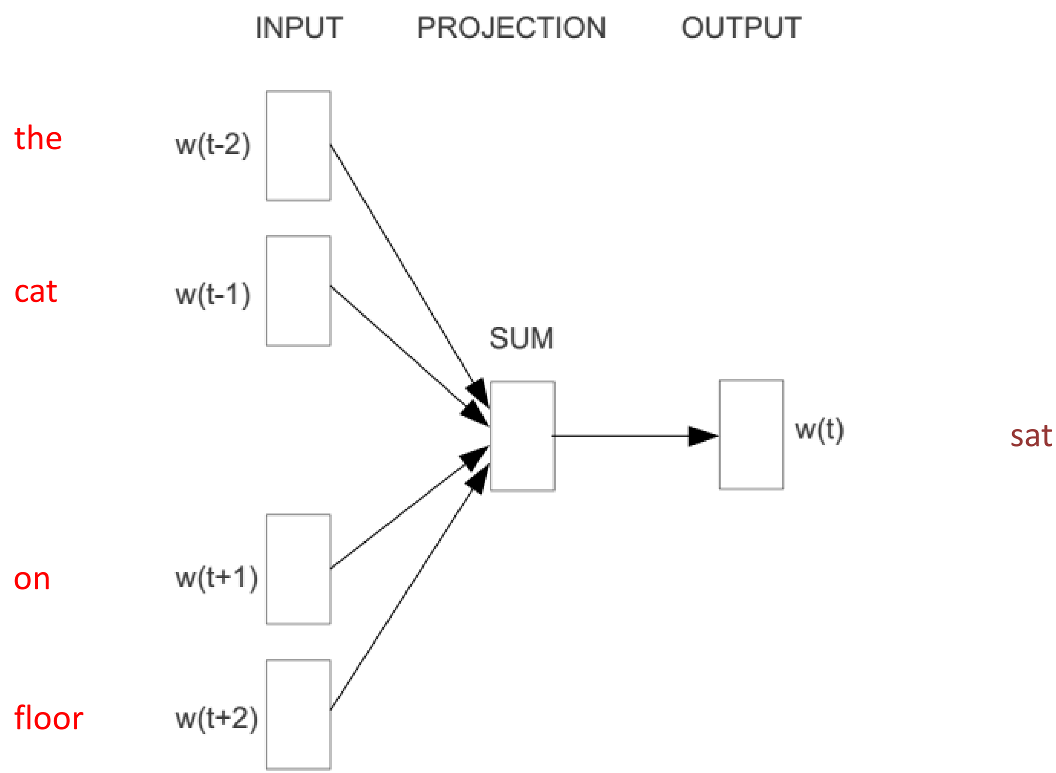
2. Skipgram: P(Context|Word)

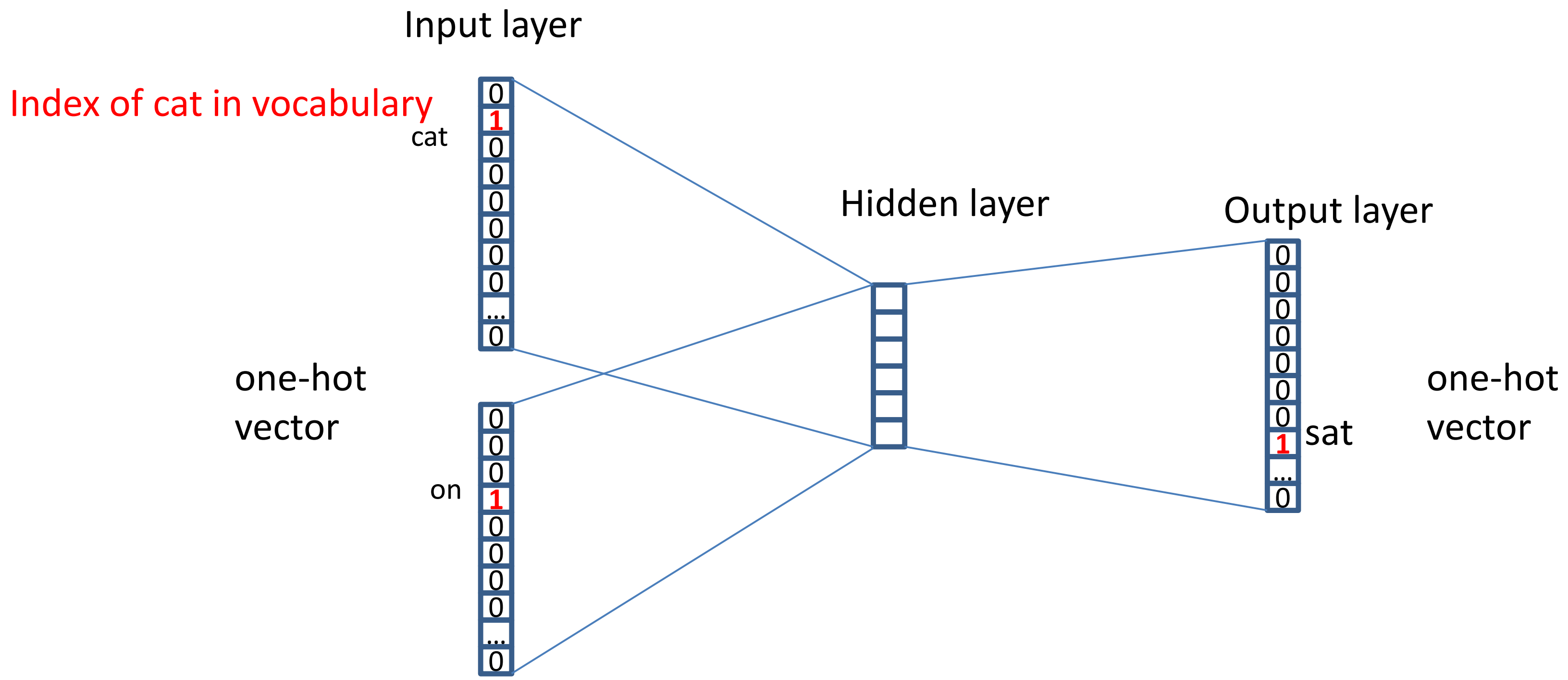




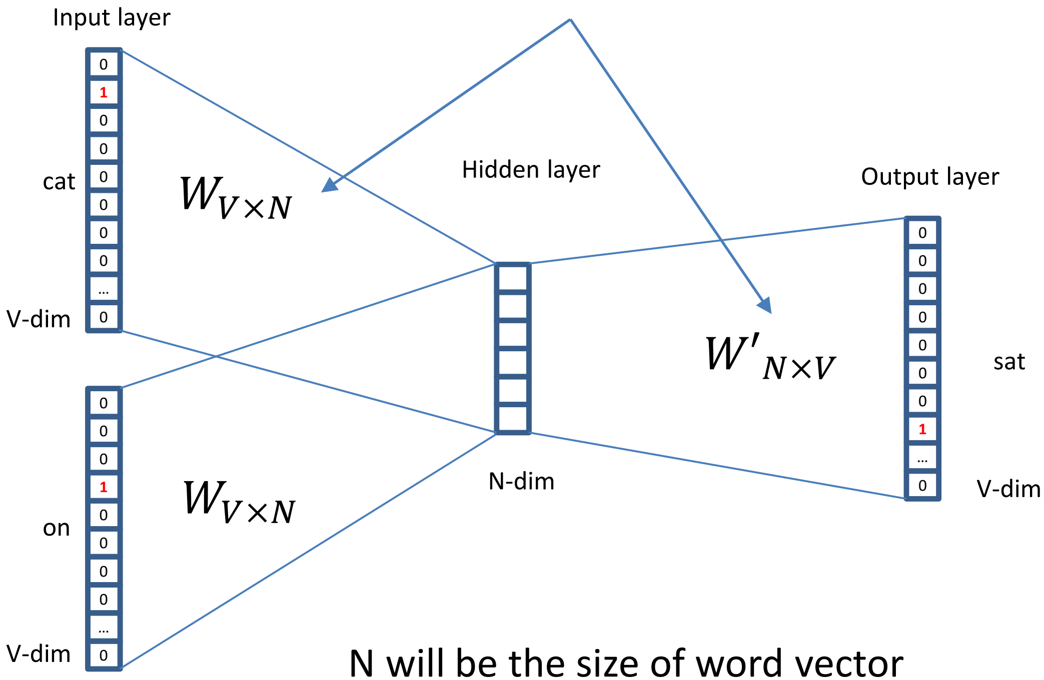
Word2vec – Continuous Bag of Word

- E.g. “The cat sat on floor”
  - Window size = 2

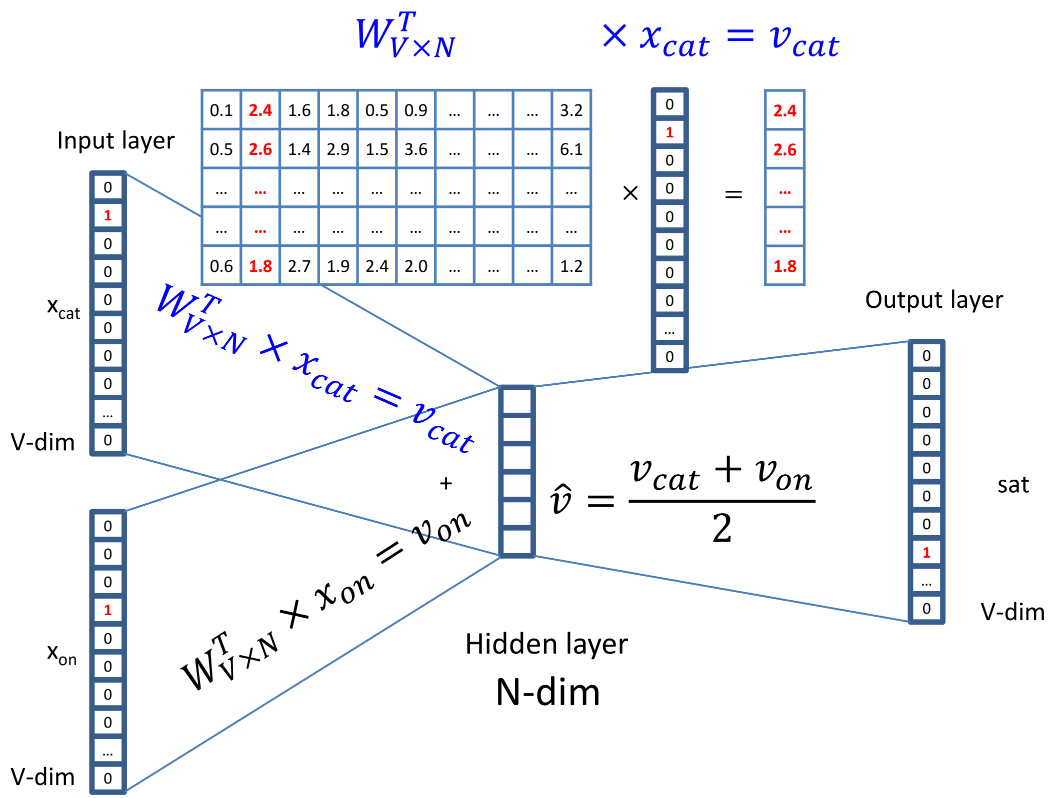



Input layer
Index of cat in vocabulary
0
1
cat
0
0
0
0
0
0
...
0
Hidden layer
Output layer
one-hot
vector
0
0
0
1
on
0
0
0
0
...
0
0
0
0
0
0
0
0
1
sat
...
0
one-hot
vector

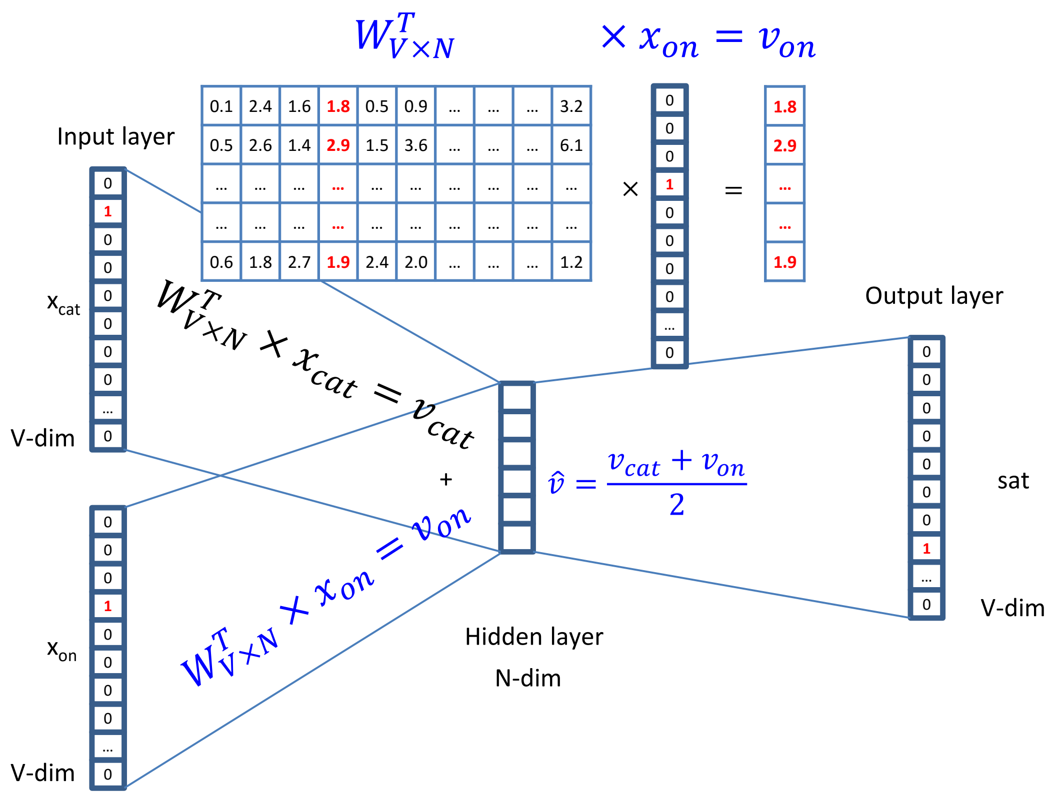
We must learn W and W'



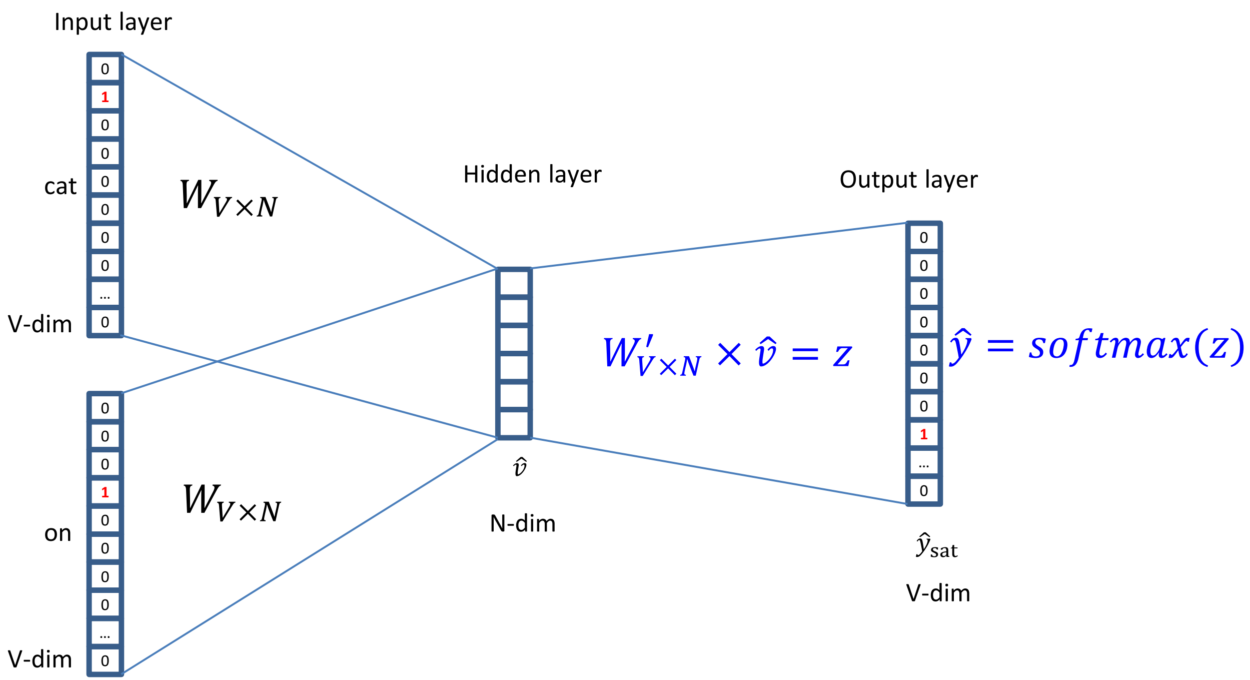
N will be the size of word vector



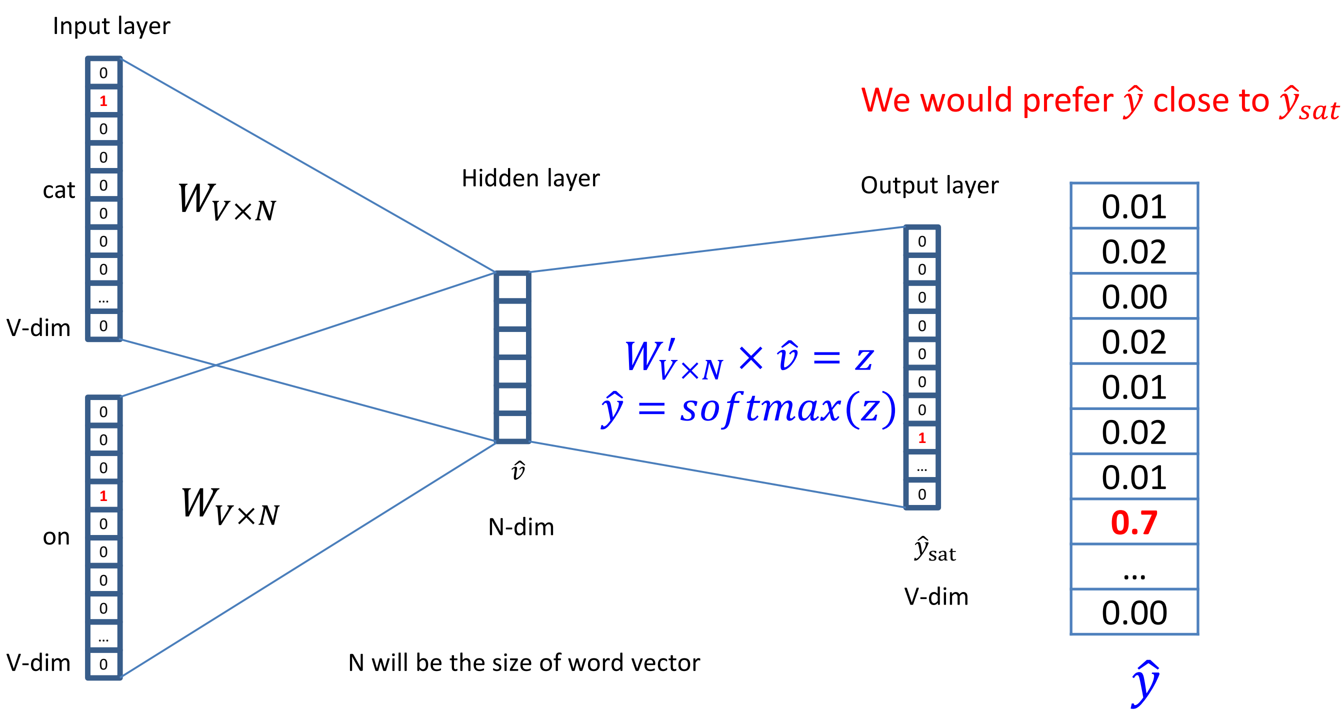
$W^T_{V\times N}$ $\times x_{cat} = v_{cat}$
Input layer
x_{cat}
V-dim
x_{on}
V-dim
$W^T_{V\times N} \times x_{cat} = v_{cat}$
+
$W^T_{V\times N} \times x_{on} = v_{on}$
$\hat{v} = \frac{v_{cat} + v_{on}}{2}$
Hidden layer
N-dim
Output layer
sat
V-dim



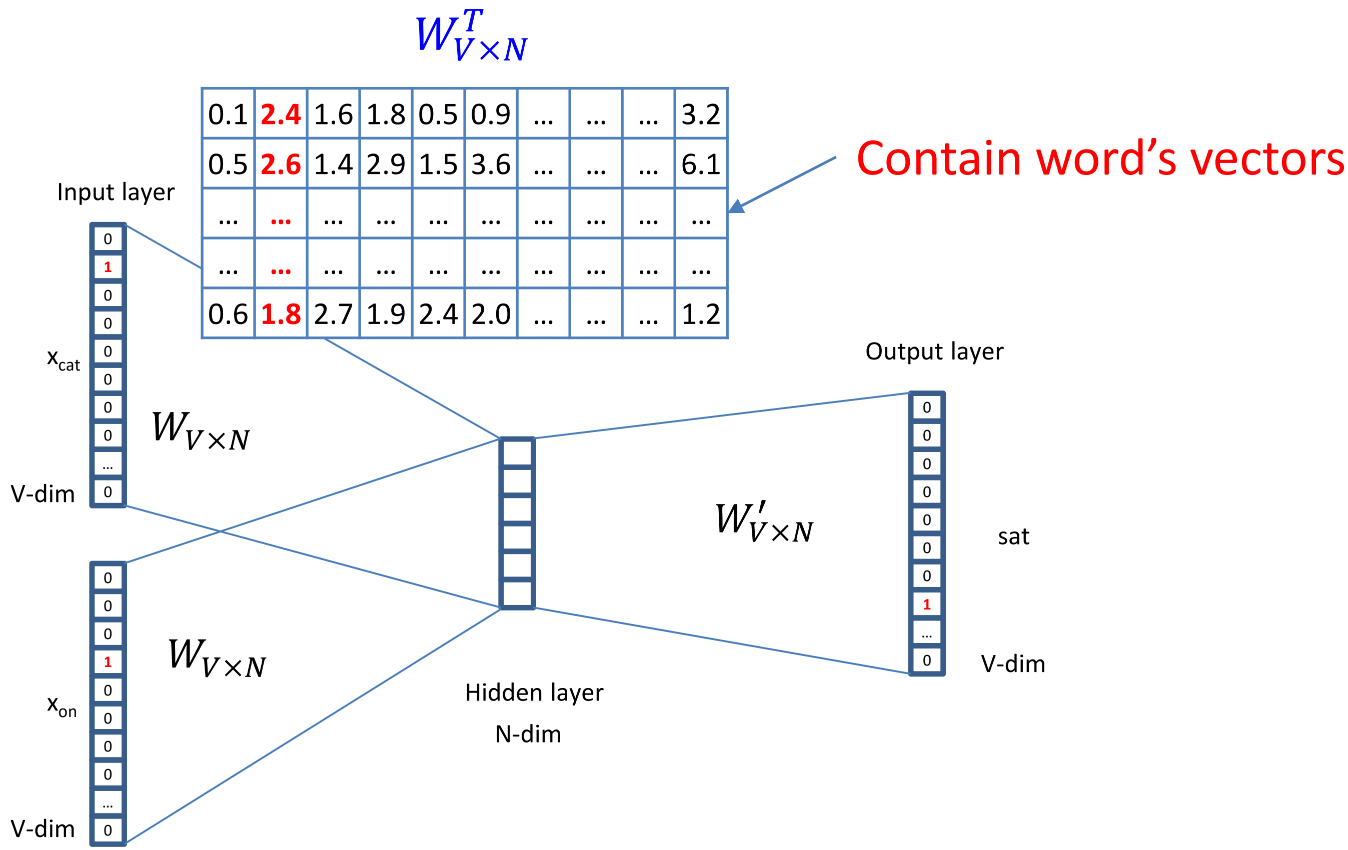
$W_{V\times N}^{T}$
$\times x_{on} = v_{on}$
Input layer
Output layer
x_{cat}
V-dim
$W_{V\times N}^{T} \times x_{cat} = v_{cat}$
+
$W_{V\times N}^{T} \times x_{on} = v_{on}$
$\hat{v} = \frac{v_{cat} + v_{on}}{2}$
sat
x_{on}
V-dim
V-dim
Hidden layer
N-dim



Input layer
cat
V-dim
on
V-dim
0
1
0
0
0
0
0
0
...
0
0
0
0
1
0
0
0
0
...
0
$W_{V\times N}$
$W_{V\times N}$
Hidden layer
$\hat{v}$
N-dim
$W'_{V\times N} \times \hat{v} = z$
Output layer
0
0
0
0
0
0
0
1
...
0
$\hat{y} = softmax(z)$
$\hat{y}_{\text{sat}}$
V-dim



Input layer
cat
V-dim
on
V-dim
$W_{V\times N}$
$W_{V\times N}$
Hidden layer
$\hat{v}$
N-dim
Output layer
$\hat{y}_{sat}$
V-dim
$W'_{V\times N} \times \hat{v} = z$
$\hat{y} = softmax(z)$
We would prefer $\hat{y}$ close to $\hat{y}_{sat}$
0.01
0.02
0.00
0.02
0.01
0.02
0.01
0.7
...
0.00
$\hat{y}$
N will be the size of word vector



0.1	2.4	1.6	1.8	0.5	0.9	…	…	…	3.2
0.5	2.6	1.4	2.9	1.5	3.6	…	…	…	6.1
…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…
0.6	1.8	2.7	1.9	2.4	2.0	…	…	…	1.2

We can consider either W or W' as the word's representation. Or even take the average.

Some interesting results

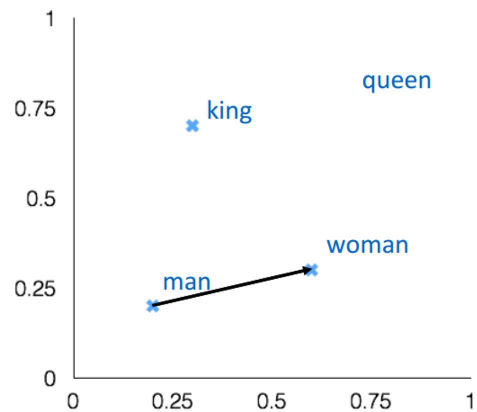
Word Analogies

Test for linear relationships, examined by Mikolov et al. (2014)

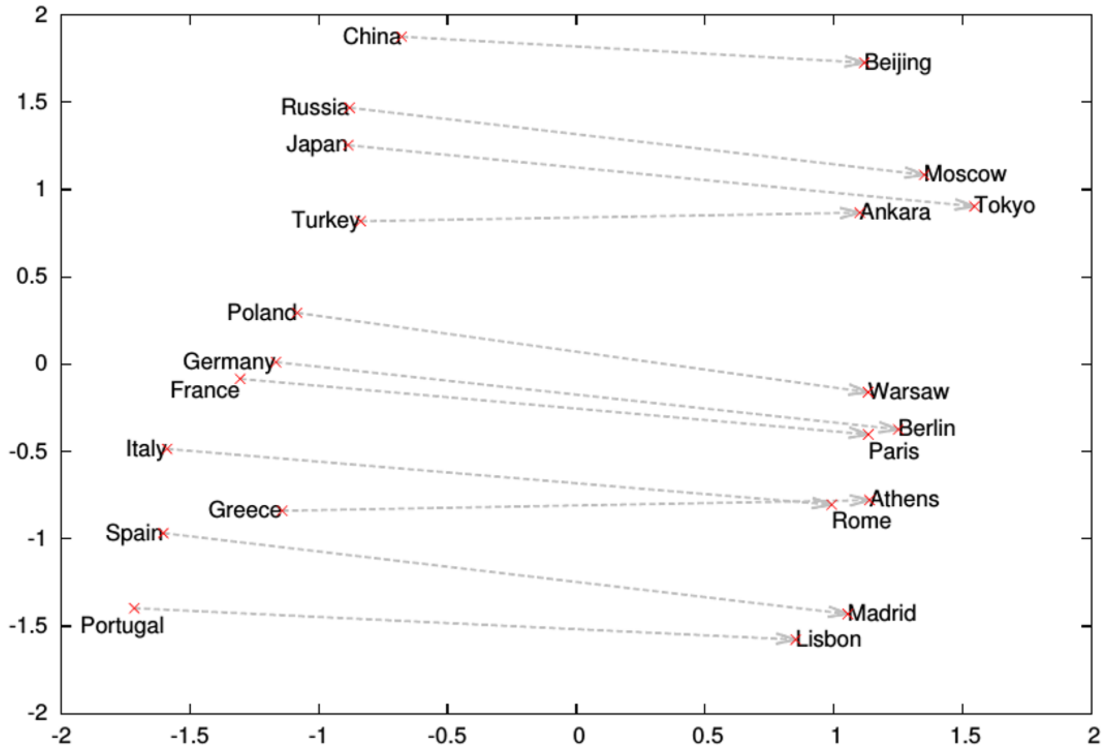
a:b :: c:? → $d = \arg\max_x \frac{(w_b - w_a + w_c)^T w_x}{||w_b - w_a + w_c||}$

man:woman :: king:?

+	king	[0.30 0.70]
-	man	[0.20 0.20]
+	woman	[0.60 0.30]
	queen	[0.70 0.80]



Word analogies



Word2Vec Objective

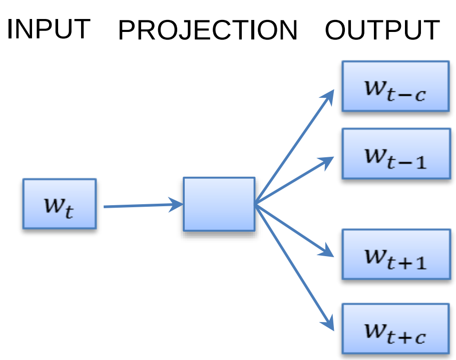
$$J(\theta) = \frac{1}{T} \sum_{t=1}^{T} \sum_{-m \leq j \leq m, j \neq 0} \log p(w_j | c)$$

Word2Vec Objective

$$p(w_j | c) = \frac{\exp(w_j^T c)}{\sum_{i=1}^{N} \exp(w_i^T c)}$$

Skipgram Model

- Input: Central word w_t
- Output: Words in its context: $\boldsymbol{w}_{\mathbf{con}}$ $\{w_{t-c}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+c}\}$
- Each input word represented by a 1-hot encoding of size V



Source Text:

Deep Learning attempts to learn multiple levels of representation from data.

Input output pairs :

Positive samples:

(representation, levels)

(representation, of)

(representation, from)

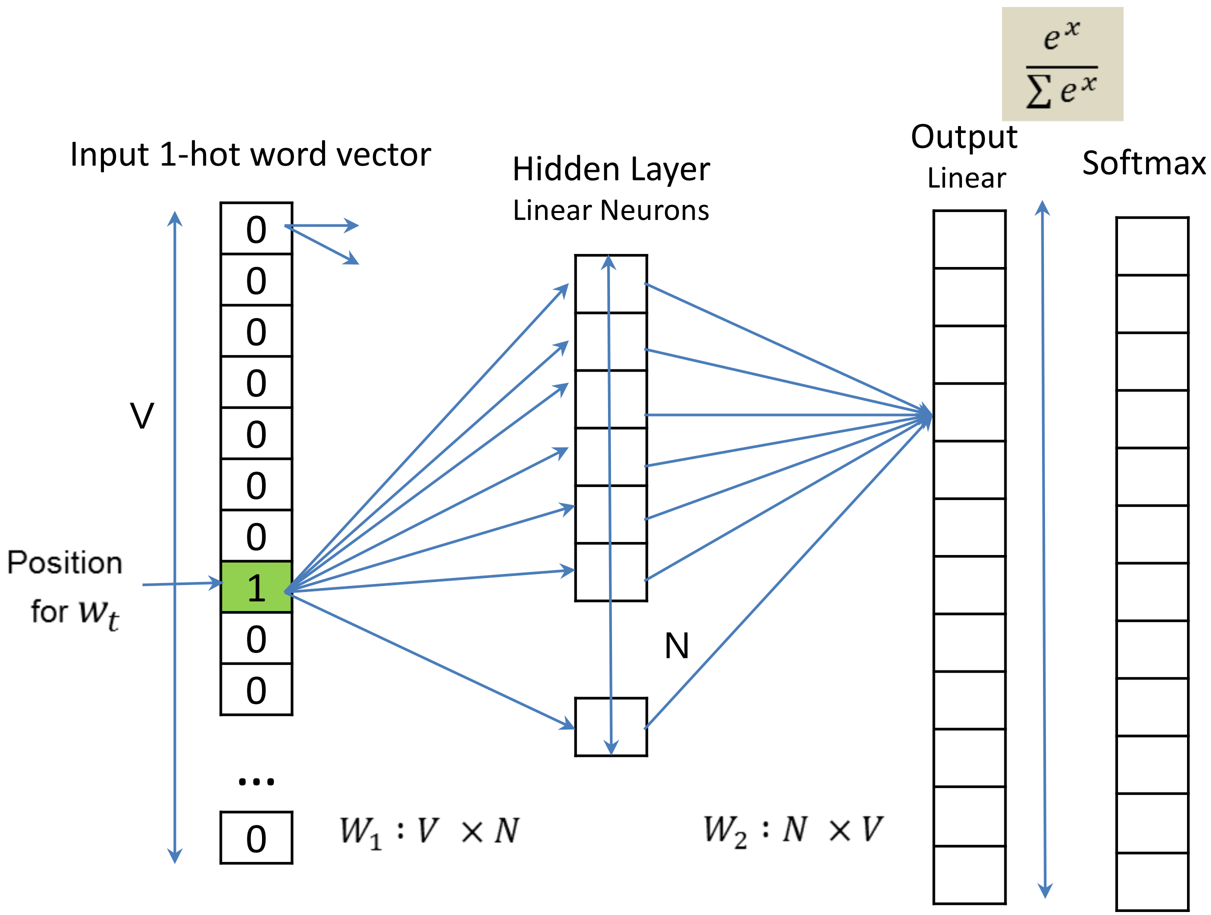
(representation, data)

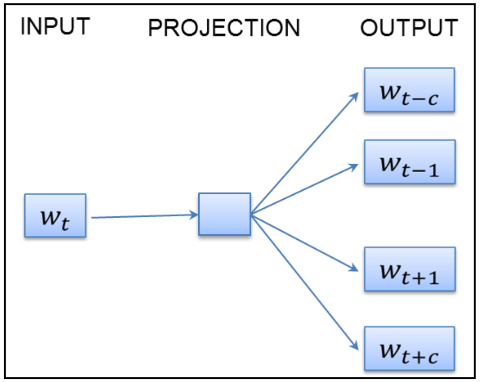
Negative samples:

(representation, x)

[x: all other words except the 4 positive]

Skipgram Model





Probability that the word in a context position Is w_i

Positive sampling
Negative sampling

Skipgram: Loss function

- Maximize $$\frac{1}{T}\sum_{t=1}^{T}\sum_{\text{context}} \log p(w_{\text{context}}|w_t)$$

$p(w_{\text{con}}|w_t)$ is the output of softmax classifier

$$p(w_{Ccon}|w_t) = \frac{\exp\left(v'_{w_{con}} \cdot v_{wt}\right)}{\sum_{W=1}^{W} exp\left(v'_{W} \cdot v_{wt}\right)}$$

Let the model parameters be θ. The solution is given by

$$\underset{\theta}{argmax} \sum_{(w_t, w_C)\in D} \log p(w_{con}|w_t;\theta)$$

$$= \sum_{(w_t, w_C)\in D}\left(\log e^{(v'_{w_{con}} \cdot v_{wt})} - \log \sum_{x} e^{(v'_{x} \cdot v_{wt})}\right)$$

Time O(V)

V: vocabulary size

Improve Efficiency

1. Hierarchical softmax: $O(\log V)$

Skipgram: Loss function

- Maximize $\frac{1}{T}\sum_{t=1}^{T}\sum_{\text{context}} \log p(w_{\text{context}}|w_t)$

$p(w_{\text{con}}|w_t)$ is the output of softmax classifier

$$p(w_{Ccon}|w_t) = \frac{\exp\left(v'_{w_{con}} \cdot v_{wt}\right)}{\sum_{W=1}^{W} exp(v'_{W} \cdot v_{wt})}$$

Let the model parameters be θ. The solution is given by

$$\underset{\theta}{argmax} \sum_{(w_t,w_C)\in D} \log p(w_{con}|w_t;\theta)$$

$$= \sum_{(w_t,w_C)\in D} \left(\log e^{(v'_{w_{con}} \cdot v_{wt})} - \log \sum_{x} e^{(v'_{x} \cdot v_{wt})} \right)$$

2. Negative sampling: Sample instead of taking all contexts into account

$$\sum_{(w_t,w_C)\in D} \left(\log \sigma(v'_{w_{con}} \cdot v_{wt}) + \sum_{i=1}^{k} \mathbb{E}_{w_i\sim P_n(w)}\left[\log \sigma(-v'_{w_i} \cdot v_{w_t})\right] \right)$$

Subsampling of frequent words

Skip-Grams with Negative Sampling (SGNS)

Marco saw a furry little wampimuk hiding in the tree.

"word2vec Explained…"
Goldberg & Levy, arXiv 2014

Skip-Grams with Negative Sampling (SGNS)

Marco saw a furry little wampimuk hiding in the tree.

words	**contexts**
wampimuk	furry
wampimuk	little
wampimuk	hiding
wampimuk	in
…	…

D (data)

"word2vec Explained…"
Goldberg & Levy, arXiv
2014

Skip-Grams with Negative Sampling (SGNS)

Maximize: $\sigma(\vec{w} \cdot \vec{c})$

- c was **observed** with w

words	**contexts**
wampimuk	furry
wampimuk	little
wampimuk	hiding
wampimuk	in

Minimize: $\sigma(\vec{w} \cdot \vec{c}\,')$

- c' was **hallucinated** with w

words	**contexts**
wampimuk	Australia
wampimuk	cyber
wampimuk	the
wampimuk	1985

"word2vec Explained…" Goldberg & Levy, arXiv 2014