Indian Institute of Technology Kharagpur
Department of Computer Science & Engineering

CS60075
Natural Language Processing
Autumn 2020

Lecture 2C : Word Representation

Sep 11 2020

Sudeshna Sarkar

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. Singular Value Decomposition (SVD) on co-occurrence matrix

Drawbacks:

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

Supervised learning

- Input: training set

$$\{(x_i, y_i)\}, (x_i, y_i) \sim D(X \times Y)$$

- Output (probabilistic model):

$$f: X \to Y$$

$$\underset{y}{argmax}\ p(y|x)$$

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

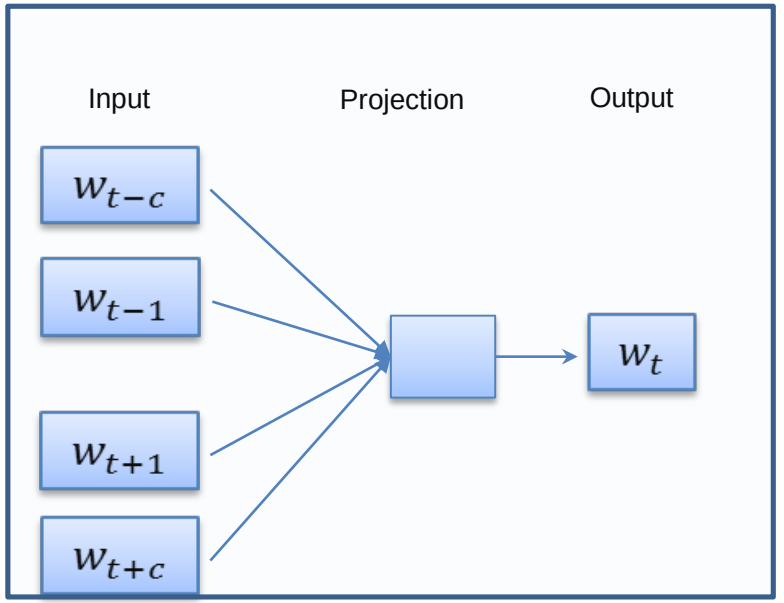
Word2vec

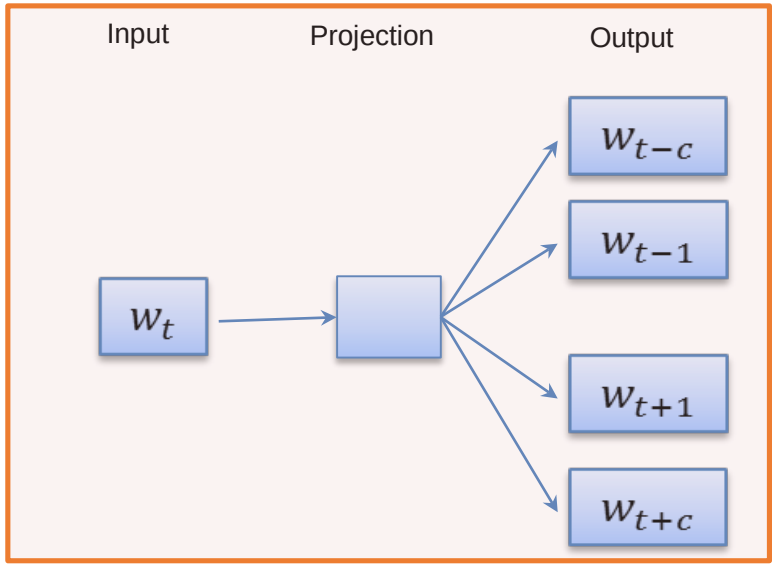
- Representation of words

``Similar words have similar contexts”

1. CBOW: P(Word|Context)

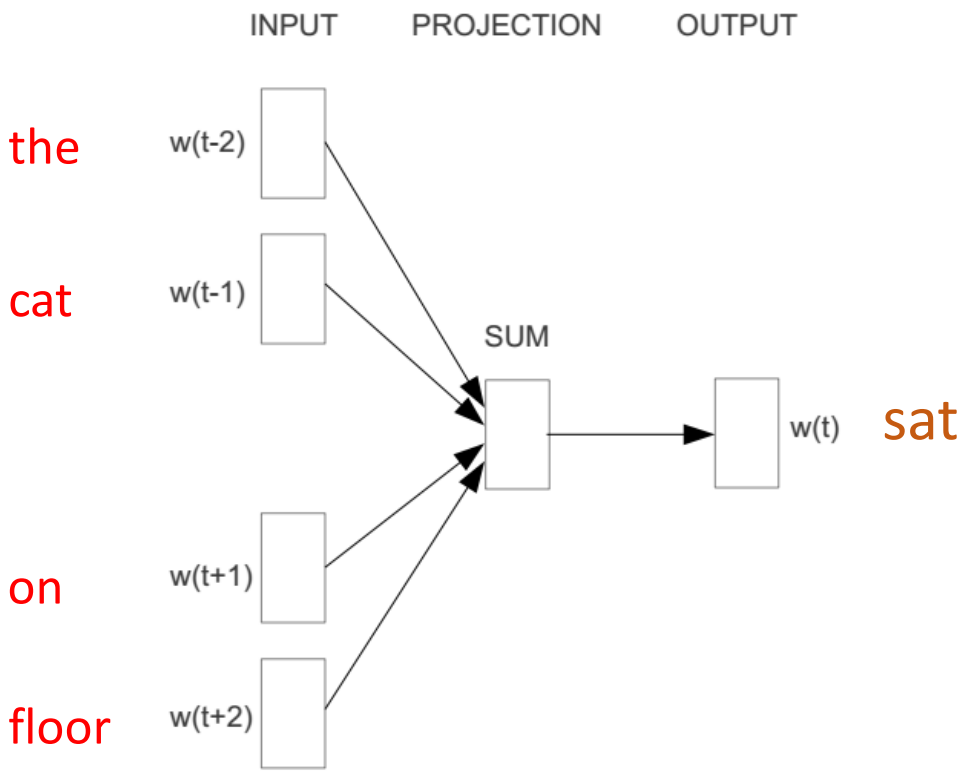
2. Skipgram: P(Context|Word)





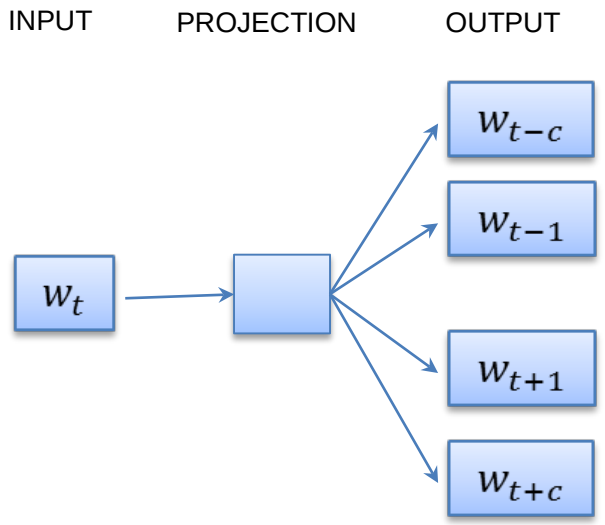
Word2vec – Continuous Bag of Word

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



Skipgram Model

- Input: Central word w_t
- Output: Words in its context: $\boldsymbol{w_{con}}$
  $\{w_{t-c}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+c}\}$
- 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)

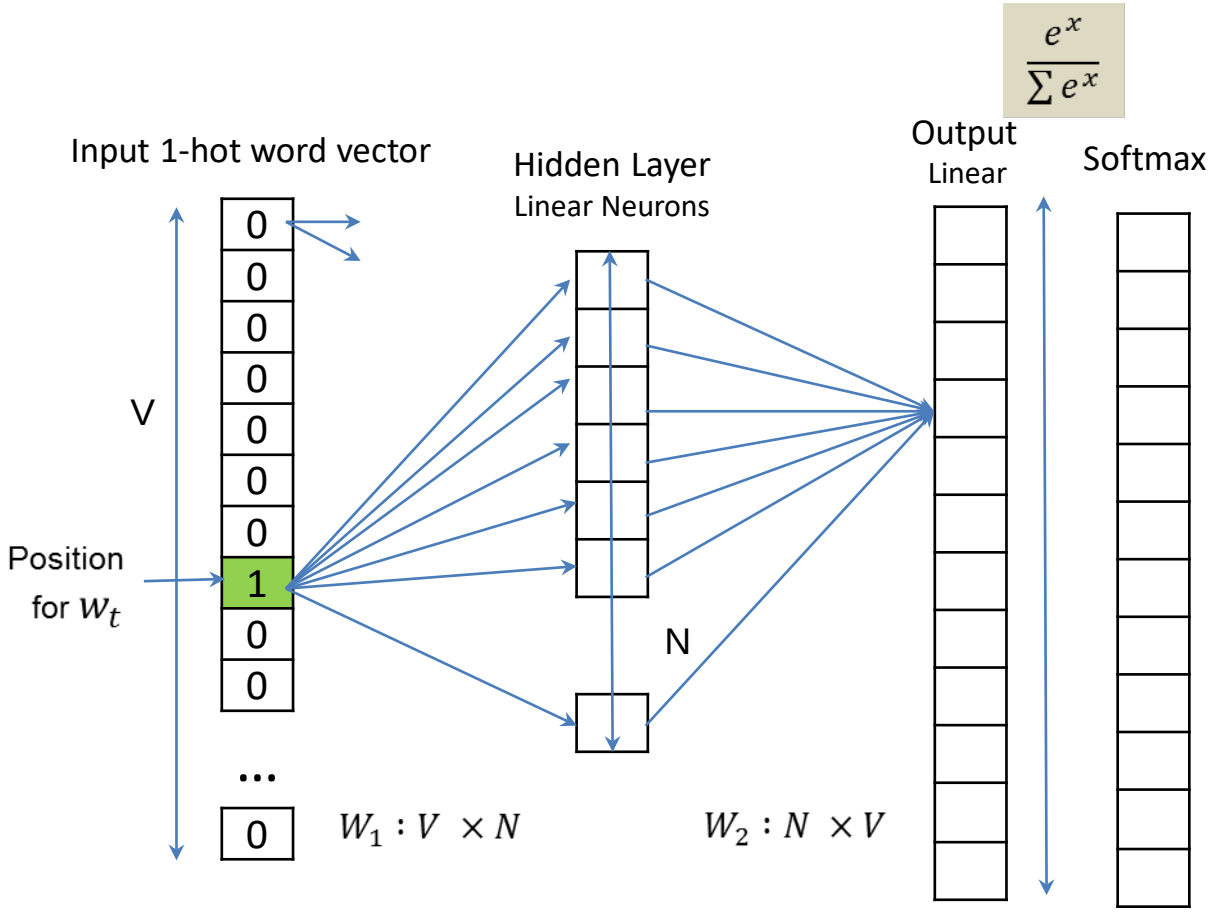
(representation, data)

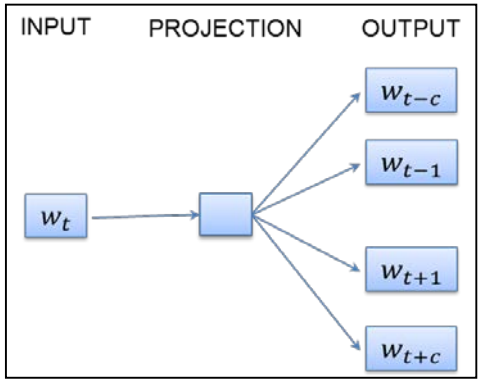
Negative samples:

(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

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

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

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

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

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

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

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

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

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

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

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

Subsampling of frequent words

Skip-Grams with Negative Sampling (SGNS)

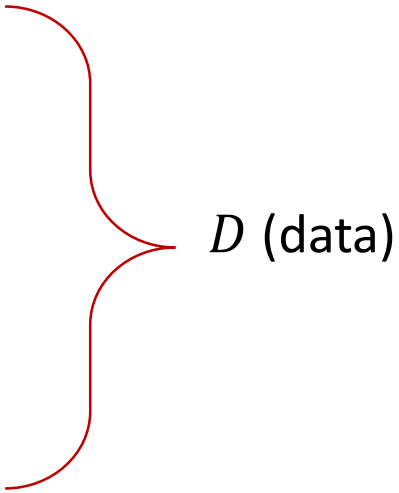
Marco saw a furry little wampimuk hiding in the tree.

"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
…	…



"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