Week 13
Word Distributed Representation

Wang Maoquan

Department of Computer Science, East China Normal University

December 5, 2016

Outline

- Word Vector
- Distributional Representation
  - Distributional Semantic Models (Matrix Factorization)
  - Distributed Representation (Neural Netword)
- Measurement
- Conclusion

Word Vector

One-hot Representation

- without semantic information
- curse of dimensionality

Distributional Hypothesis

words that occur in the same contexts tend to have similar meanings (Harris, 1954)

Distributed Representation

Mapping words to K-dimensionality vector space
Learning distributed representations of concepts [Hinton, 1986]

Distributional Representation

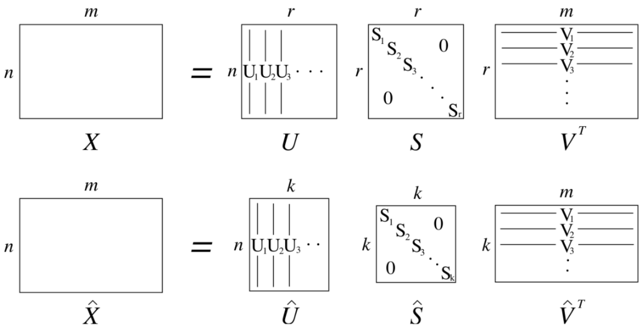
- Distributional Semantic Models (Matrix Factorization)
- Distributed Representation (Neural Netword)

Window based Co-occurence Matrix

- I like deep learning.
- I like NLP.
- I enjoy flying.

counts	I	like	enjoy	deep	learning	NLP	flying	.
I	0	2	1	0	0	0	0	0
like	2	0	0	1	0	1	0	0
enjoy	1	0	0	0	0	0	1	0
deep	0	1	0	0	1	0	0	0
learning	0	0	0	1	0	0	0	1
NLP	0	1	0	0	0	0	0	1
flying	0	0	1	0	0	0	0	1
.	0	0	0	0	1	1	1	0

SVD of Co-occurence Matrix



- $\hat{X}$ is the best rank k approximation to X, in terms of least squares

Distributional Semantic Models

1. Choose of context (word-document, word-word, word-Ngram)
2. element value of Matrix (The co-occurrence frequency)
3. (optional) Matrix Factorization (SVD, NMF, CCA, HPCA)

Language Model

Language Model Unified Definition

$$p(s) = p(w_1, w_2, ..., w_n) = \prod_{t=1}^{T} p(w_i | Context) \quad (1)$$

1. n-gram

$$p(s) = \prod_{t=1}^{T} p(w_t | w_{t-1}, ..., w_{t-n+2}, w_{t-n+1}) \quad (2)$$

there n = 2, 3 (Bigram, Trigram)

2. n-pos

$$p(s) = p(s) = \prod_{t=1}^{T} p(w_t | c(w_{t-n+1}), c(w_{t-n+2}), ..., c(w_{t-1})) \quad (3)$$

Language Model

$$p(w_i|w_{i-(n-1)}, ..., w_{i-1}) = \frac{count(w_{i-(n-1),...,w_{i-1},w_i})}{count(w_{i-(n-1),...,w_{i-1}})} \quad (4)$$

cons

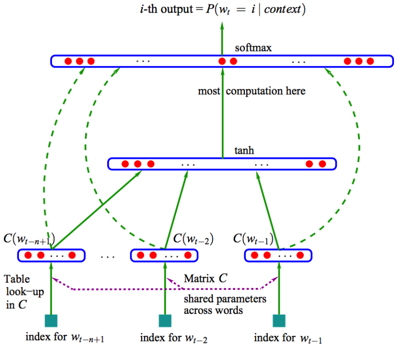
1. Choose of n
2. Long range dependent problem
3. Data Sparseness Problem → Smoothing

Distributed Representation

Models

- Neural Network Language Model (NNLM) [Bengio, 2003]
- Log-Bilinear Language Model (LBL) [Hinton, 2007]
- Recurrent Neural Network Language Model (RNNLM) [Mikolov, 2012]
- C&W [Collobert and Weston, 2008]
- Word2Vec [Tomas, 2013]
- Sentiment Embeddings [Tangdy, 2014]

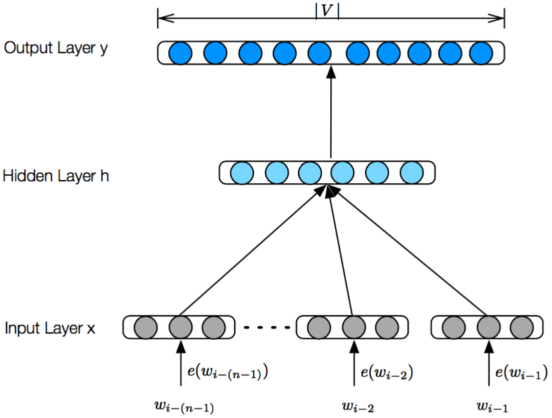
Neural Network Language Model



$$y = b + Wx + U\tanh(d + Hx) \tag{5}$$

$$P(w_t|w_{t-1}, ..., w_{t-n+1}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_{w_i}}} \tag{6}$$

Neural Network Language Model



$$P(w_i|w_{i-(n-1)}, \ldots, w_{i-1}) = \frac{e^{y(w_i)}}{\sum_{k=1}^{|V|} e^{y(v_k)}}$$

$$y = b^{(2)} + Wx + Uh$$

$$h = tanh(b^{(1)} + Hx)$$

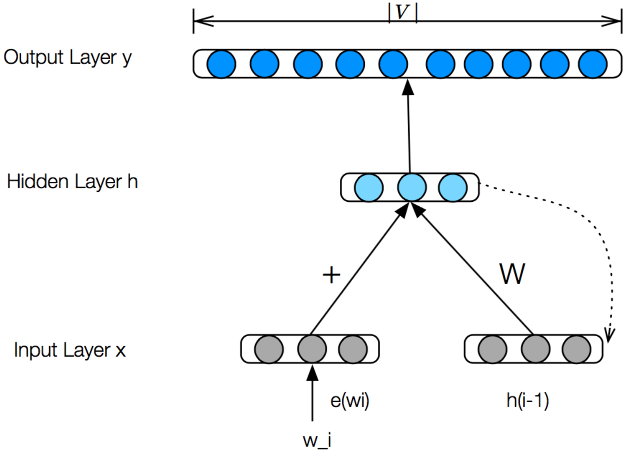
$$x = [e(w_{i-(n-1)}); \ldots; e(w_{i-2}); e(w_{i-1})]$$

Log-Bilinear Language Model

Three New Graphical Models for Statistical Language Modelling [Mnih,Hinton, 2007]

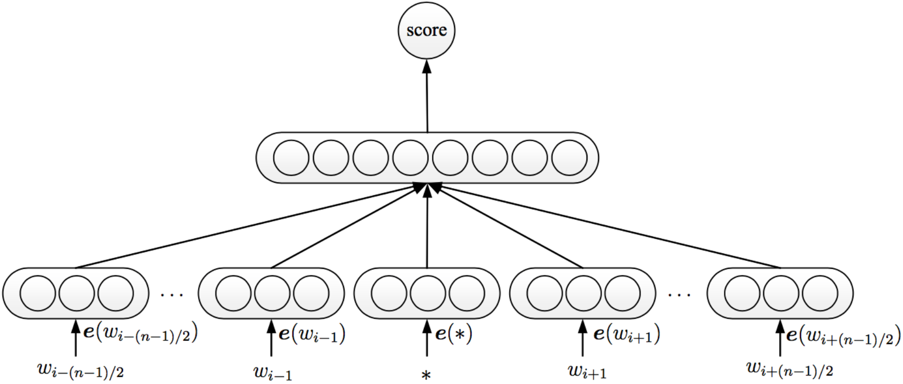
$$y(w_i) = b^2 + e(w_i)^{\mathrm{T}} b^1 + e(w_i)^{\mathrm{T}} H[e(w_{i-(n-1)}); ...; e(w_{i-2}); e(w_{i-1})] \quad (7)$$

Recurrent Neural Network based Language Model



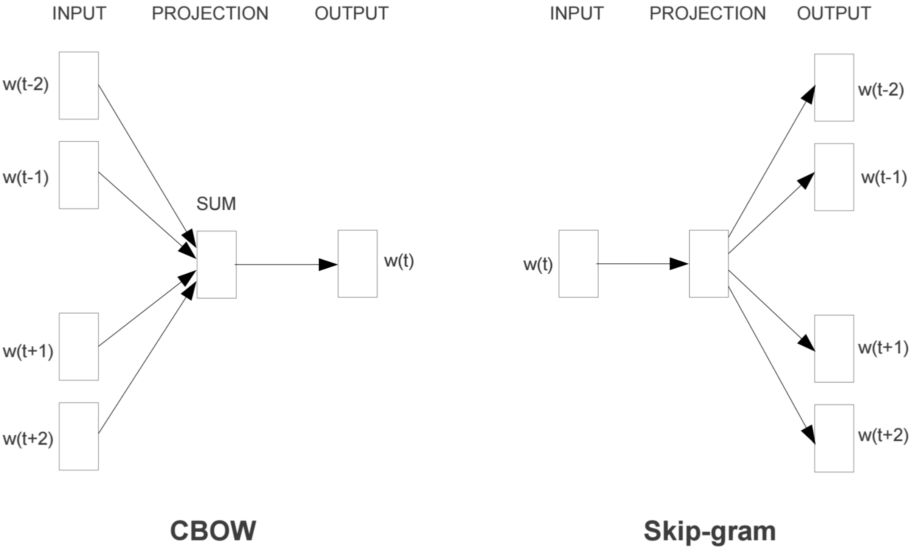
$$h(i) = \phi(e(w_i) + Wh(i-1)) \tag{8}$$

C&W



(9)

Word2Vec


INPUT
PROJECTION
OUTPUT
w(t-2)
w(t-1)
SUM
w(t)
w(t+1)
w(t+2)
CBOW
INPUT
PROJECTION
OUTPUT
w(t)
w(t-2)
w(t-1)
w(t+1)
w(t+2)
Skip-gram

Measurement

1. linguistic characteristics (word similarity, tfl, sem, syn)
2. use word vector as features (avg for classification, named entity recognition)
3. input as other neural networks (CNN, RNN, LSTM, MNT)

Conclusion

Model	Relation of w, c	Representation of c
Skip-gram	c predicts w	one of c
CBOW	c predicts w	average of c
Order	c predicts w	concatenation
LBL	c predicts w	compositionality
NNLM	c predicts w	compositionality
C&W	scores w, c	compositionality

简单

复杂