# Word Embedding

Changzhi Sun

East China Normal University

*changzhisun@stu.ecnu.edu.cn*

October 26, 2016

# Overview

1. Introduction
2. Traditional Embedding Models
3. Multiview Embedding

# Word Vector Representation

## One-hot Representation

- motel: $[0, 1, 0, 0]$
- hotel: $[0, 0, 0, 1]$

# Word Vector Representation

## One-hot Representation

- motel: $[0, 1, 0, 0]$
- hotel: $[0, 0, 0, 1]$

1. curse of dimensionality
2. similarity

# Word Vector Representation

## One-hot Representation

- motel: $[0, 1, 0, 0]$
- hotel: $[0, 0, 0, 1]$

1. curse of dimensionality
2. similarity

## Distributed Representation

- motel: $[0.856, -0.732, 0.119, 0.503]$
- hotel: $[0.744, -0.621, 0.008, 0.392]$

# Word Vector Representation

## Core

- utilize contextual information for representation
  - similar context should have similar representation

# Word Vector Representation

## Core

- utilize contextual information for representation
  - similar context should have similar representation
- 2 options: full document vs windows
  - word-document co-occurrence matrix
    - will give general topics
  - word-word co-occurrence matrix
    - capture both syntactic and semantic information
    - window length (more common: 5-10)
    - symmetric (irrelevant whether left or right context)

# Word Vector Representation

## Core

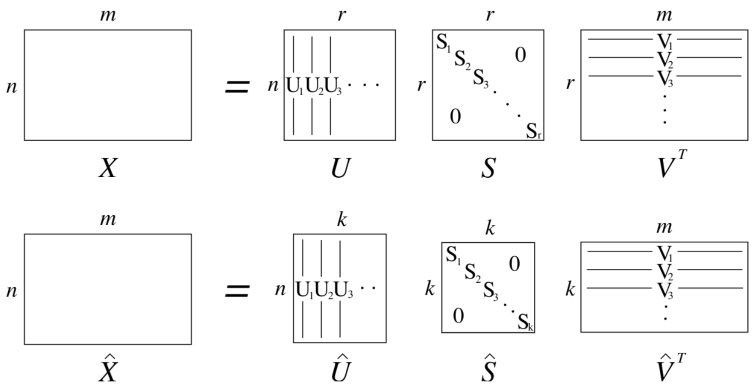
- utilize contextual information for representation
  - similar context should have similar representation
- 2 options: full document vs windows
  - word-document co-occurrence matrix
    - will give general topics
  - word-word co-occurrence matrix
    - capture both syntactic and semantic information
    - window length (more common: 5-10)
    - symmetric (irrelevant whether left or right context)
- matrix factorization (SVD) for dimensionality reduction

# 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

# Hacks to X

## Hacks

- problems: function words (the, he, has) are too frequent $\rightarrow$ syntax has too much impact. Some fixes:
  - $min(X, t)$ with $t \sim 100$
  - ignore them all
- ramped windows that count closer words more
- use Pearson correlations instead of counts, then set negative values to 0
- +++

# Directly Learn Word Vector

## Models

- Skip-gram
- CBOW
- NNLM
- C&W
- GloVe
- ...

# Skip-gram(word2vec)

## Main idea

- instead of capturing co-occurrence counts directly
- predict surrounding words of every word (window length)

# Skip-gram(word2vec)

## Main idea

- instead of capturing co-occurrence counts directly
- predict surrounding words of every word (window length)
- fast and can easily incorporate a new sentence or add a word to the vocab

# Skip-gram(word2vec)

## Main idea

- instead of capturing co-occurrence counts directly
- predict surrounding words of every word (window length)
- fast and can easily incorporate a new sentence or add a word to the vocab
- object function: maximize the log probability of any context word given the current center word

# Skip-gram(word2vec)

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

$$p(w_O|w_I) = \frac{exp(v'^{T}_{w_O} v_{w_I})}{\sum_{w=1}^{W} exp(v'^{T}_{w} v_{w_I})}$$

- v and v' are **input** and **output** vector representation of w
- every word has two vectors
- **dynamic** logistic regression

# Derivations of Gradient

$$\frac{exp(u_O^T v_C)}{\sum_{w=1}^{W} exp(u_w^T v_C)}$$

# Derivations of Gradient

$$\frac{exp(u_O^T v_C)}{\sum_{w=1}^{W} exp(u_w^T v_C)}$$

$$\frac{\partial J}{\partial v_C} = u_O - \frac{\partial \ log \sum_{w=1}^{W} exp(u_w^T v_C)}{\partial v_C}$$

# Derivations of Gradient

$$\frac{exp(u_O^T v_C)}{\sum_{w=1}^{W} exp(u_w^T v_C)}$$

$$\frac{\partial J}{\partial v_C} = u_O - \frac{\partial \ log \sum_{w=1}^{W} exp(u_w^T v_C)}{\partial v_C}$$

$$= u_O - \frac{1}{\sum_{w=1}^{W} exp(u_w^T v_C)} \sum_{x=1}^{W} (exp(u_x^T v_C) u_x)$$

# Derivations of Gradient

$$\frac{exp(u_O^T v_C)}{\sum_{w=1}^{W} exp(u_w^T v_C)}$$

$$\begin{aligned}
\frac{\partial J}{\partial v_C} &= u_O - \frac{\partial \ log \sum_{w=1}^{W} exp(u_w^T v_C)}{\partial v_C} \\
&= u_O - \frac{1}{\sum_{w=1}^{W} exp(u_w^T v_C)} \sum_{x=1}^{W} (exp(u_x^T v_C) u_x) \\
&= u_O - \sum_{x=1}^{W} (\frac{exp(u_x^T v_C)}{\sum_{w=1}^{W} exp(u_w^T v_C)} u_x)
\end{aligned}$$

# Derivations of Gradient

$$\frac{exp(u_O^T v_C)}{\sum_{w=1}^{W} exp(u_w^T v_C)}$$

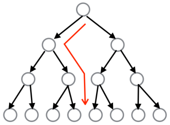
$$\begin{aligned}
\frac{\partial J}{\partial v_C} &= u_O - \frac{\partial \ log \sum_{w=1}^{W} exp(u_w^T v_C)}{\partial v_C} \\
&= u_O - \frac{1}{\sum_{w=1}^{W} exp(u_w^T v_C)} \sum_{x=1}^{W} (exp(u_x^T v_C) u_x) \\
&= u_O - \sum_{x=1}^{W} \left(\frac{exp(u_x^T v_C)}{\sum_{w=1}^{W} exp(u_w^T v_C)} u_x\right) \\
&= u_O - \sum_{x=1}^{W} (P(w_x | w_C) u_x)
\end{aligned}$$

# Skip-gram(word2vec)

1. large vocabularies
   - train too slowly and not scalable

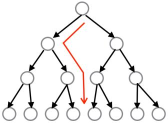
# Skip-gram(word2vec)

1. large vocabularies
   - train too slowly and not scalable
2. hierarchical softmax (WordNet)



# Skip-gram(word2vec)

1. large vocabularies
   - train too slowly and not scalable
2. hierarchical softmax (WordNet)



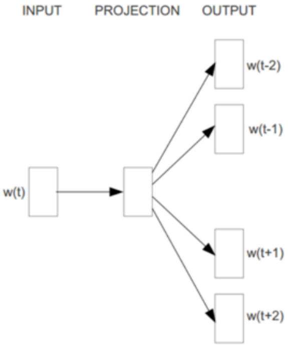
3. negative sampling

$$log\ \sigma({v'_{w_O}}^T v_{w_I}) + \sum_{i=1}^{k} \mathbf{E}_{w_i \sim P_n(w)}[log\ \sigma(-{v'_{w_i}}^T v_{w_I})]$$

# Skip-gram(word2vec)

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{-c\le j\le c, j\ne 0} P(w_{i+j} \mid w_i)$$

$$P(w_i \mid w_j) = \frac{\exp(v_i'^T v_j)}{\sum_{w_i} \exp(v_i'^T v_j)}$$



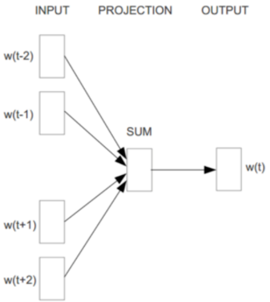
# CBOW

- Continued Bag of Words Model

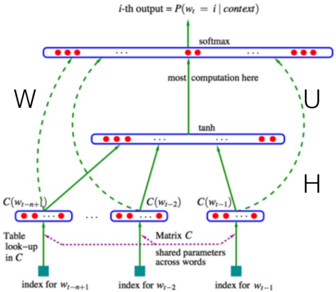
$$\frac{1}{N}\sum_{i=1}^{N} P(w_i \mid w_{i-k}, w_{i-k+1}, \cdots, w_{i-1}, w_{i+1}, \cdots, w_{i+k-1}, w_{i+k})$$

$$P(w_i \mid C_i) = \frac{\exp(v_i'^{T} v_{C_i})}{\sum_{w_i} \exp(v_i'^{T} v_{C_i})}$$

$$v_{C_i} = \sum_{j \in C_i} v_j$$



- Neural Network Language Model

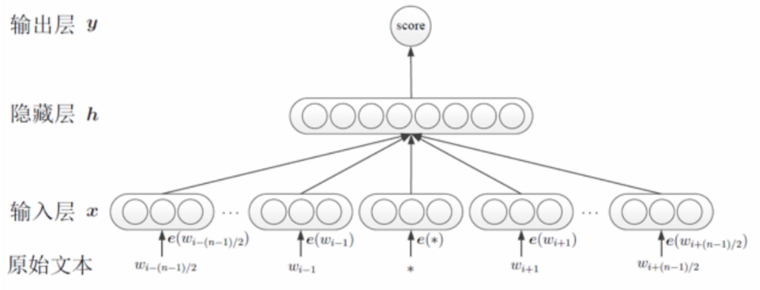



$$y = b + Wx + U\tanh(d + Hx)$$

$$x = (C(w_{t-1}), C(w_{t-2}), \cdots, C(w_{t-n+1}))$$

$$\theta \leftarrow \theta + \varepsilon \frac{\partial \log \hat{P}(w_t | w_{t-1}, \cdots w_{t-n+1})}{\partial \theta}$$

# C&W




目标函数 $$\max(0, 1 - s(w, c) + s(w', c))$$

# GloVe

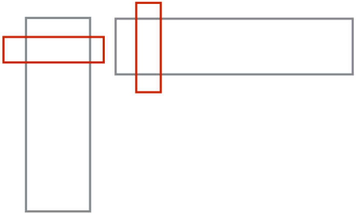
- Global Vector

$$J = \sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2$$

词向量　　词向量　　词词共现

| | w1 | w2 | w3 | w4 |
|---|---|---|---|---|
| w1 | | 2 | 4 | 1 |
| w2 | 2 | | 3 | |
| w3 | 4 | 3 | | 1 |
| w4 | 1 | | 1 | |

$\approx$



# Evaluation Task

1. analogy task
   - syn: predict - predicting $\approx$ dance - dancing
   - sem: king - queen $\approx$ man - woman
2. similarity task
   - $\rho_{X,Y} = \frac{Cov(X,Y)}{\sigma_X \sigma_Y}$
3. tfl
4. sentiment classification
5. NER, POS tag

# Model 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

# Conclusion

1. no perfect match, only match perfect
    - using domain corpus training

# Conclusion

1. no perfect match, only match perfect
   - using domain corpus training
2. more corpus better performance
   - small corpus $\rightarrow$ simple model (Skip-gram)
   - huge corpus $\rightarrow$ complex model (NNLM, C&W)

# Conclusion

1. no perfect match, only match perfect
   - using domain corpus training
2. more corpus better performance
   - small corpus → simple model (Skip-gram)
   - huge corpus → complex model (NNLM, C&W)
3. domain more important than data size

# Conclusion

1. no perfect match, only match perfect
   - using domain corpus training
2. more corpus better performance
   - small corpus $\rightarrow$ simple model (Skip-gram)
   - huge corpus $\rightarrow$ complex model (NNLM, C&W)
3. domain more important than data size
4. embedding size $> 50$

# Conclusion

1. no perfect match, only match perfect
   - using domain corpus training
2. more corpus better performance
   - small corpus $\rightarrow$ simple model (Skip-gram)
   - huge corpus $\rightarrow$ complex model (NNLM, C&W)
3. domain more important than data size
4. embedding size $> 50$
5. regularization
   - weight decay, early stop, drop out

# Conclusion

1. no perfect match, only match perfect
    - using domain corpus training
2. more corpus better performance
    - small corpus $\rightarrow$ simple model (Skip-gram)
    - huge corpus $\rightarrow$ complex model (NNLM, C&W)
3. domain more important than data size
4. embedding size $> 50$
5. regularization
    - weight decay, early stop, drop out
6. other tricks
    - batch normalization
    - momentum

# Learning Multiview Embeddings of Twitter Users

## Motivation & Contributions

- capture information from all aspects of user's online life
  - their tweets
  - tweets of mentioned users
  - friends
  - followers
- GCCA can learn a vector from each of views
- betten than concatenating views into a single vector
- evaluation

# Generalized Canonical Correlation Analysis

## GCCA

$$\underset{G,U_i}{argmin} \sum_i ||G - X_i U_i||_F^2 \quad s.t. G'G = I$$

- $X_i \in \mathcal{R}^{n \times d_i}$, data matrix for the $i$th view
- $U_i \in \mathcal{R}^{d_i \times k}$, mapping matrix
- $G \in \mathcal{R}^{n \times k}$, user representation

## weighted GCCA

$$\underset{G,U_i}{argmin} \sum_i w_i ||G - X_i U_i||_F^2 \quad s.t. G'G = I, w_i \geq 0$$

# Expriments

1. show the performance of multiview embeddings compared to other representations, not on building the best system
2. tasks
    - User Engagement Prediction
        - determine which topics a user will likely tweet about
    - Friend Recommendation
        - recommend other accounts for a user to follow
    - Demographic Characteristics Inference
        - binary superviesed prediction task
        - old/young, male/female republican/democrat

# Conclusions

1. proposed several representations of Twitter users
2. multiview approach that combines these views into a single embedding
3. achieve promising results on three different prediction tasks
4. learning user representation(kernel PCA, Deep CCA, multitask DL)

# References I


- Adrian Benton, Raman Arora, and Mark Dredze, *Learning multiview embeddings of twitter users*, Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (Berlin, Germany), Association for Computational Linguistics, August 2016, pp. 14–19.
- Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin, *A neural probabilistic language model*, journal of machine learning research **3** (2003), no. Feb, 1137–1155.
- Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa, *Natural language processing (almost) from scratch*, Journal of Machine Learning Research **12** (2011), no. Aug, 2493–2537.

# References II


- Sergey Ioffe and Christian Szegedy, *Batch normalization: Accelerating deep network training by reducing internal covariate shift*, arXiv preprint arXiv:1502.03167 (2015).
- Omer Levy and Yoav Goldberg, *Neural word embedding as implicit matrix factorization*, Advances in neural information processing systems, 2014, pp. 2177–2185.
- Omer Levy, Yoav Goldberg, and Ido Dagan, *Improving distributional similarity with lessons learned from word embeddings*, Transactions of the Association for Computational Linguistics **3** (2015), 211–225.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, *Efficient estimation of word representations in vector space*, arXiv preprint arXiv:1301.3781 (2013).

# References III


- Andriy Mnih and Koray Kavukcuoglu, *Learning word embeddings efficiently with noise-contrastive estimation*, Advances in Neural Information Processing Systems, 2013, pp. 2265–2273.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean, *Distributed representations of words and phrases and their compositionality*, Advances in neural information processing systems, 2013, pp. 3111–3119.
- Jeffrey Pennington, Richard Socher, and Christopher D Manning, *Glove: Global vectors for word representation.*, EMNLP, vol. 14, 2014, pp. 1532–43.

# The End