Word2Vec

Presenters:
Akash Sanjay Mehta
Jaithrik Yadav Bollaboina
Parin Rajesh Sanghavi
Priyaranjan Behera
Sai Sri Harsha Kunapareddy

Word2Vec

- Word2vec is a group of related models that are used to produce word embeddings.
- Pioneered by Google. Two papers published by Mikolov et al. (2013)
- Shallow, two-layer neural networks, that are trained to retain linguistic contexts of words.
- Two Models:
  - Skip-Gram
  - CBOW (Continuous Bag of Words)

Atomic and Contextual Word Representation

Atomic Identity Representation:

Apple - 0 0 0 0 0 0 0 1 0 0 0 0 0

Orange - 0 0 0 0 0 1 0 0 0 0 0 0 0

Car - 0 0 0 0 0 0 0 0 0 1 0 0 0

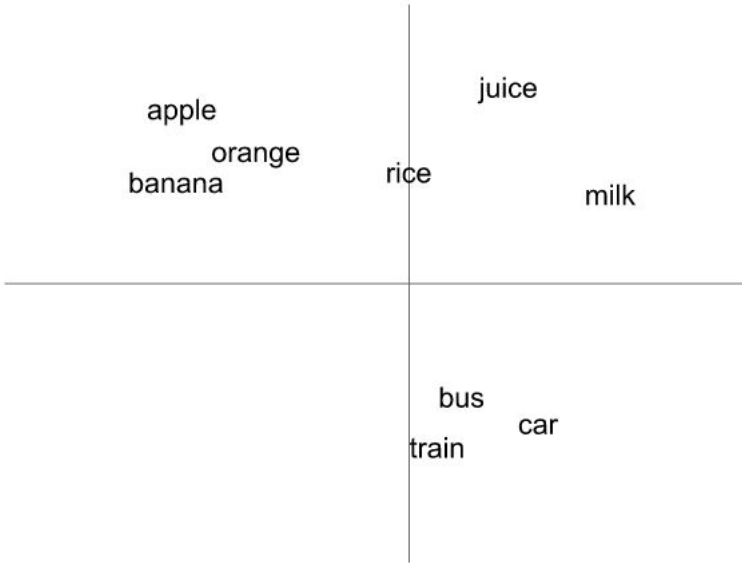
Contextual Representation:

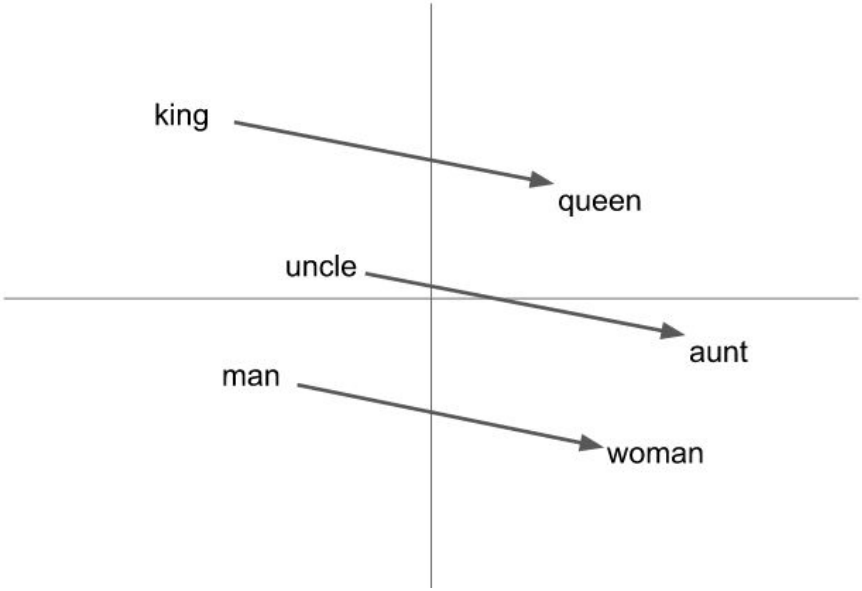
I eat an apple everyday.

I eat an orange everyday.

I like driving my car to work.

Word Vectors

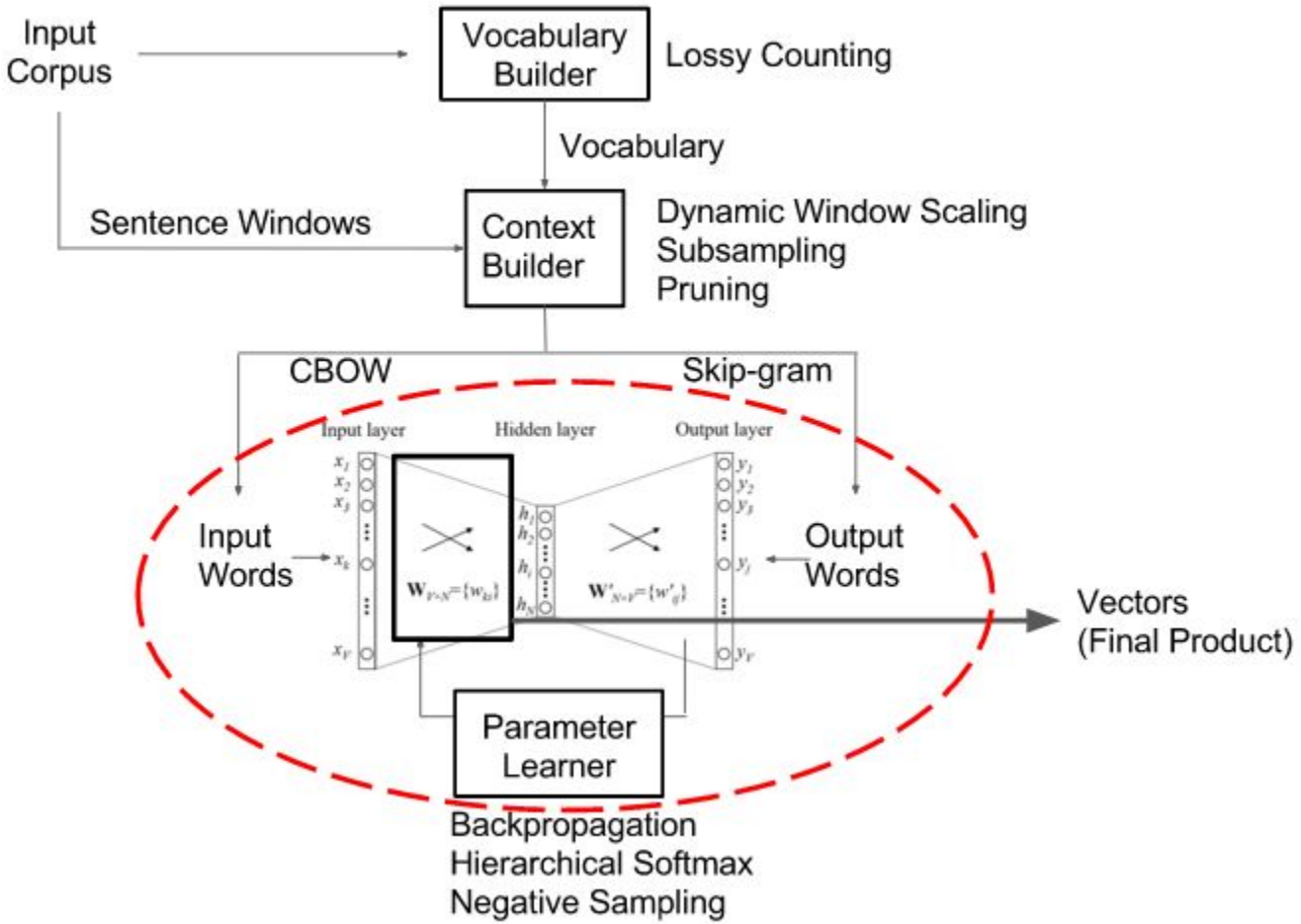




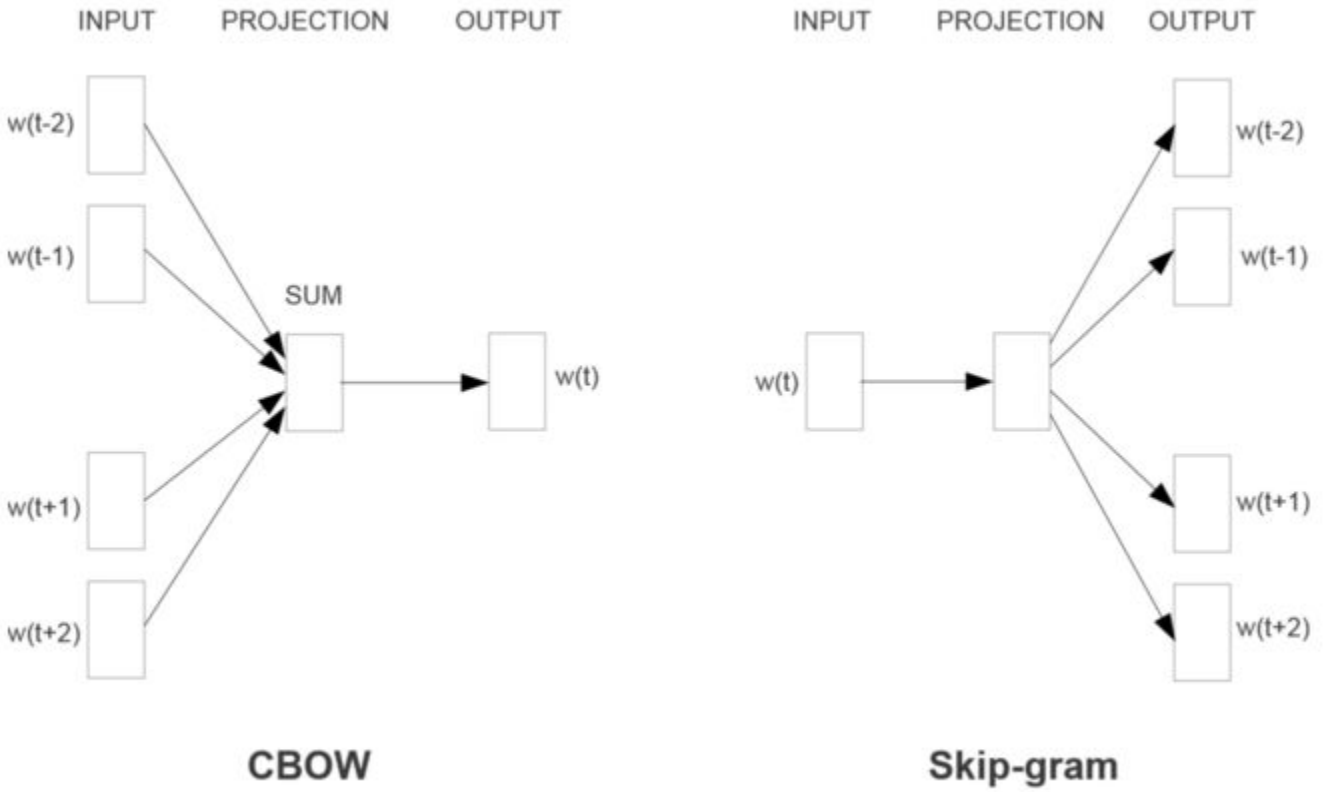
Use of Word Embeddings

- Dependency Parsing
- Named Entity Recognition
- Document Classification
- Sentiment Analysis
- Paraphrase Detection
- Word Clustering
- Machine Learning Translation

Word2Vec Process Flow



Two Models: CBOW and Skip-Gram

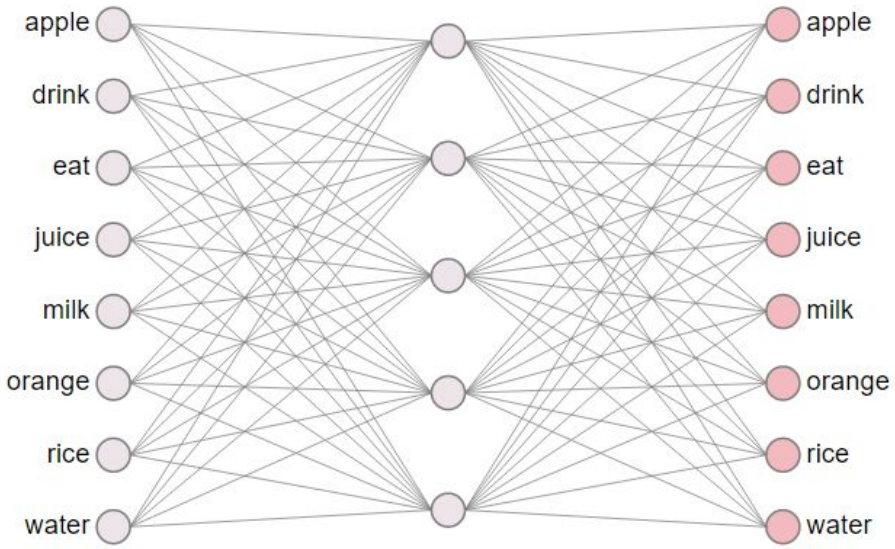


Inputs and Outputs

Method 1: continuous bag-of-word (CBOW)

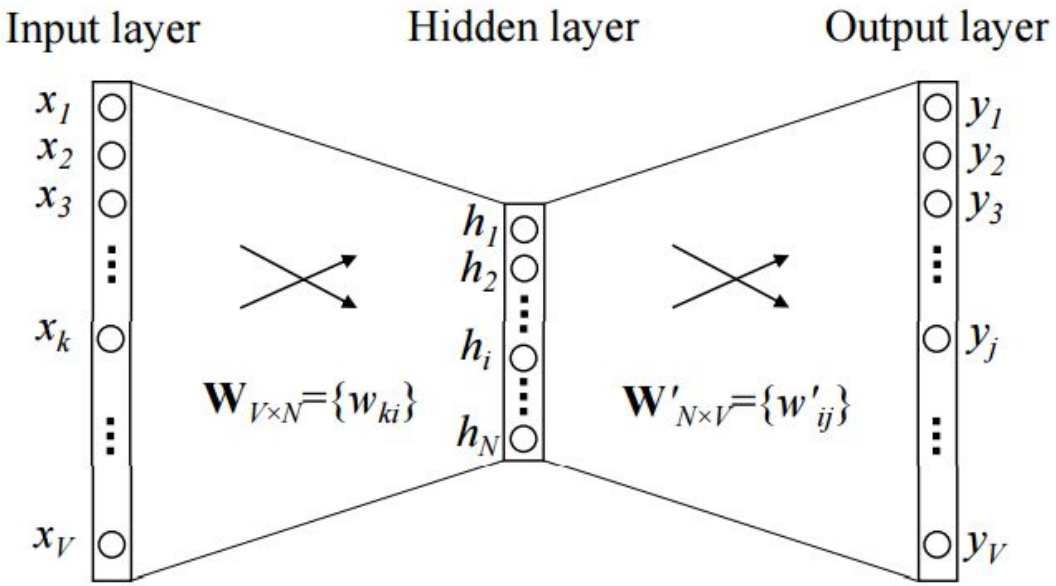
Method 2: skip-gram (SG)

Structure Highlights



- Input Layer:
  - One-Hot Vector
- Hidden Layer:
  - Linear Activation Function
- Output Layer:
  - Softmax

Word2Vec Network

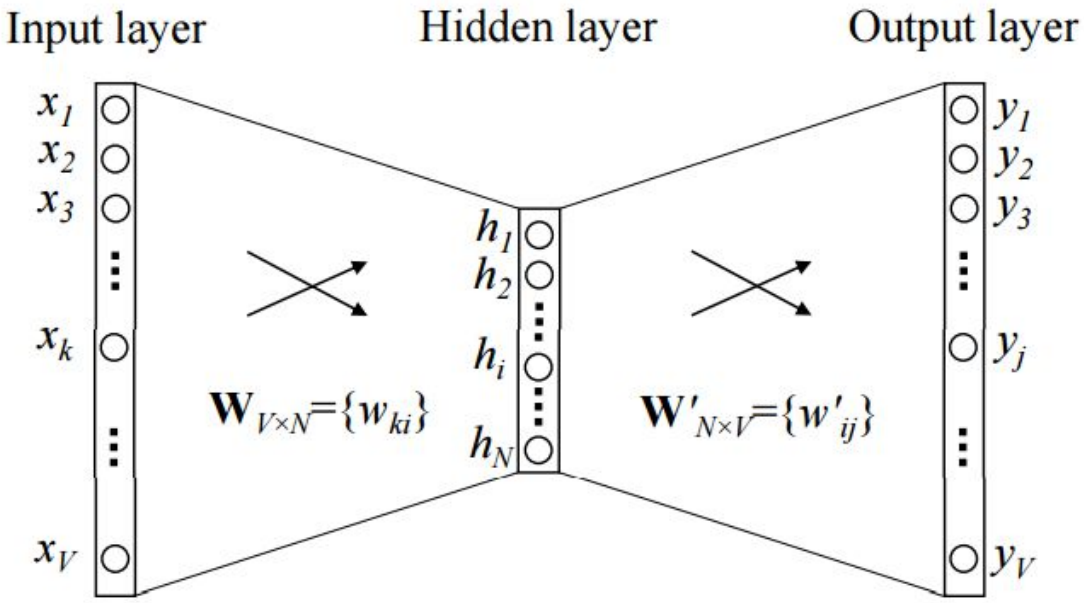


$$\mathbf{h} = \mathbf{x}^T\mathbf{W} := \mathbf{v}_{w_I}$$

$$u_j = \mathbf{v}'_{w_j}{}^T \cdot \mathbf{h}$$

$$p(w_j|w_I) = \frac{\exp\left(\mathbf{v}'_{w_O}{}^T \mathbf{v}_{w_I}\right)}{\sum_{j'=1}^{V} \exp\left(\mathbf{v}'_{w'_j}{}^T \mathbf{v}_{w_I}\right)}$$

Neural Network Training



$$E = -\log \frac{\exp\left({\mathbf{v}'_{w_O}}^T \mathbf{v}_{w_I}\right)}{\sum_{j'=1}^{V} \exp\left({\mathbf{v}'_{w'_j}}^T \mathbf{v}_{w_I}\right)}$$

$$\frac{\partial E}{\partial u_j} = y_j - t_j := e_j$$

$$\frac{\partial E}{\partial w'_{ij}} = \frac{\partial E}{\partial u_j} \cdot \frac{u_j}{\partial w'_{ij}}$$

$$\frac{\partial E}{\partial h_i} = \sum_{j=1}^{V} \frac{\partial E}{\partial u_j} \cdot \frac{\partial u_j}{\partial h_i}$$

$$\frac{\partial E}{\partial w_{ki}} = \frac{\partial E}{\partial h_i} \cdot \frac{\partial h_i}{\partial w_{ki}}$$

Word Co-occurrence

	Apple	Orange	Eat	Juice	Milk	Rice	Water
Apple	0	2	5	3	0	0	1
Orange		0	4	4	0	0	0
Eat			0	0	1	5	0
Juice				0	0	1	1
Milk					0	3	1
Rice						0	3
Water							0

Optimizations and Limitations

Optimizations to enable training:

- Hierarchical Softmax
- Negative Sampling

Limitations:

- Word Ambiguity - Multiple Meanings
- Debuggability - Black-box structure
- Handling Sequence

http://bit.ly/wevi-online

Demo

References

- Efficient Estimation of Word Representations in Vector Space - Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean
- Distributed Representations of Words and Phrases and their Compositionality - Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean
- Word2vec Parameter Learning Explained - Xin Rong