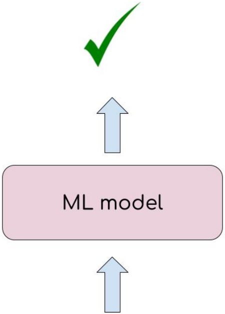
Deep Learning

14 Word Embeddings

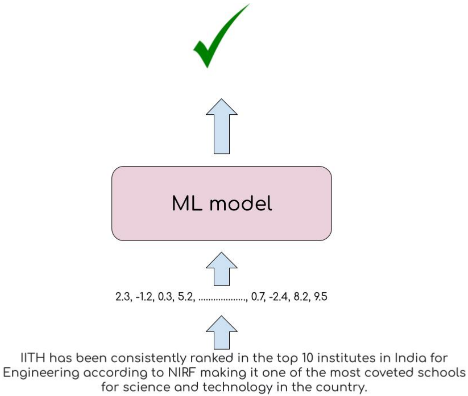
Dr. Konda Reddy Mopuri
Dept. of AI, IIT Hyderabad
Jan-May 2023

Why Word Embeddings?



IITH has been consistently ranked in the top 10 institutes in India for Engineering according to NIRF making it one of the most coveted schools for science and technology in the country.

Why Word Embeddings?



Terminology

1. Corpus: collection of authentic text organized into dataset

Terminology

1. Corpus: collection of authentic text organized into dataset
2. Vocabulary (V): Set of unique words across all the i/p streams

Terminology

1. Corpus: collection of authentic text organized into dataset
2. Vocabulary (V): Set of unique words across all the i/p streams
3. Target: Representation for every word in V

One-hot Encoding

1. $|V|$ words encoded as binary vectors of length $|V|$

Dictionary	Word Representation					
A	1	0	0	....	0	0
Bus	0	1	0	....	0	0
Cat	0	0	1	....	0	0
⋮						
Tide	0	0	0	....	1	0
Zone	0	0	0	....	0	1

One-hot encoding: Drawbacks

1. Space inefficient (e.g. 13M words in Google 1T corpus)

One-hot encoding: Drawbacks

1. Space inefficient (e.g. 13M words in Google 1T corpus)
2. No notion of similarity (or, distance) between words

Distributed Representations

1. Representation/meaning of a word should consider its context in the corpus

Distributed Representations

1. Representation/meaning of a word should consider its context in the corpus
2. **Co-occurrence matrix** can capture this!
   - size: (#words $\times$ #words)
   - rows: words (m), cols: context (n)
   - words and context can be of same or different size

Distributed Representations

1. Representation/meaning of a word should consider its context in the corpus
2. **Co-occurrence matrix** can capture this!
   - size: (#words $\times$ #words)
   - rows: words (m), cols: context (n)
   - words and context can be of same or different size
3. Context can be defined as a 'h' word neighborhood

Distributed Representations

1. Representation/meaning of a word should consider its context in the corpus
2. **Co-occurrence matrix** can capture this!
   - size: (#words $\times$ #words)
   - rows: words (m), cols: context (n)
   - words and context can be of same or different size
3. Context can be defined as a 'h' word neighborhood
4. Each row (column): vectorial representation of the word (context)

Co-occurrence matrix

$$X = \begin{array}{c} \\ I \\ like \\ enjoy \\ deep \\ learning \\ NLP \\ flying \\ . \end{array} \begin{array}{c} \begin{array}{cccccccc} I & like & enjoy & deep & learning & NLP & flying & . \end{array} \\ \begin{bmatrix} 0 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

Co-occurrence matrix

1. Very sparse

Co-occurrence matrix

1. Very sparse
2. Very high-dimensional (grows with the vocabulary size)

Co-occurrence matrix

1. Very sparse
2. Very high-dimensional (grows with the vocabulary size)
3. Solution:Dimensionality reduction (SVD)!

SVD on the Co-occurrence matrix

1. $X = U\Sigma V^T$

SVD on the Co-occurrence matrix

1. $X = U\Sigma V^T$
2. $$\begin{bmatrix} \\ X \\ \\ \end{bmatrix}_{m\times n} = \begin{bmatrix} \uparrow & \dots & \uparrow \\ u_1 & \dots & u_k \\ \downarrow & \dots & \downarrow \end{bmatrix}_{m\times k} \cdot \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_k \end{bmatrix}_{k\times k} \cdot \begin{bmatrix} \leftarrow & v_1^T & \rightarrow \\ & \vdots & \\ \leftarrow & v_k^T & \rightarrow \end{bmatrix}_{k\times n}$$

SVD on the Co-occurrence matrix

1. $X = U\Sigma V^T$
2. $$\begin{bmatrix} & & \\ & X & \\ & & \end{bmatrix}_{m\times n} =$$
$$\begin{bmatrix} \uparrow & \dots & \uparrow \\ u_1 & \dots & u_k \\ \downarrow & \dots & \downarrow \end{bmatrix}_{m\times k} \cdot \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_k \end{bmatrix}_{k\times k} \cdot \begin{bmatrix} \leftarrow & v_1^T & \rightarrow \\ & \vdots & \\ \leftarrow & v_k^T & \rightarrow \end{bmatrix}_{k\times n}$$
3. $X = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \ldots + \sigma_k u_k v_k^T$
4. $\hat{X} = \sum_{i=1}^{d<k} \sigma_i u_i v_i^T$ is a d-rank approximation of X

SVD on the Co-occurrence matrix

1. Before the SVD, representations were the rows of X

SVD on the Co-occurrence matrix

1. Before the SVD, representations were the rows of X
2. How do we reduce the representation size with SVD ?

SVD on the Co-occurrence matrix

1. Before the SVD, representations were the rows of X
2. How do we reduce the representation size with SVD ?
3. $W_{\text{word}} = U_{m \times k} \cdot \Sigma_{k \times k}$

SVD on the Co-occurrence matrix

1. $W_{\text{word}} \in \mathbb{R}^{m \times k}$ $(k \ll |V| = m)$ are considered the representation of the words

SVD on the Co-occurrence matrix

1. $W_{\text{word}} \in \mathbb{R}^{m \times k}$ $(k \ll |V| = m)$ are considered the representation of the words
2. Lesser dimensions but the same similarities! (one may verify that $XX^T = \hat{X}\hat{X}^T$)

SVD on the Co-occurrence matrix

1. $W_{\text{word}} \in \mathbb{R}^{m \times k}$ $(k \ll |V| = m)$ are considered the representation of the words
2. Lesser dimensions but the same similarities! (one may verify that $XX^T = \hat{X}\hat{X}^T$)
3. $W_{\text{context}} = V \in \mathbb{R}^{n \times k}$ are taken as the representations for the context words

Count-based vs prediction-based models

1. Techniques we have seen so far rely on the counts (or, co-occurrence of words)

Count-based vs prediction-based models

1. Techniques we have seen so far rely on the counts (or, co-occurrence of words)
2. Next, we see prediction based models for word embeddings

Word2Vec

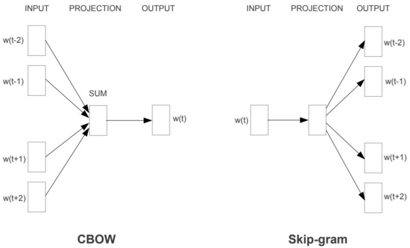
1. T Mikolov et al. (2013)

Word2Vec

1. T Mikolov et al. (2013)
2. Predict words from the context

Word2Vec

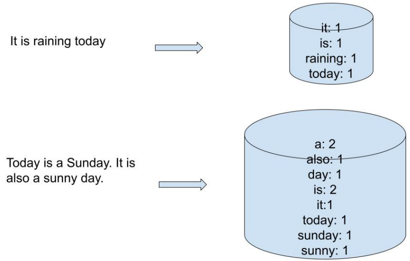
1. T Mikolov et al. (2013)
2. Predict words from the context
3. Two versions: Continuous Bag of Words (CBoW) and Skip-gram



Caption

Bag of Words (BoW)

1. Bag of Words: Collection and frequency of words

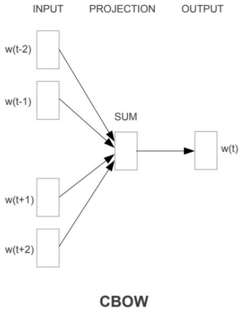


CBoW

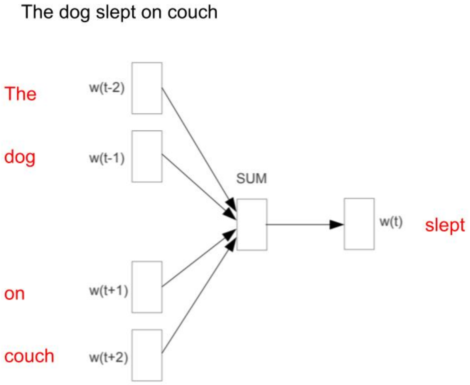
1. Considers the embeddings of ‘h’ words before and ‘h’ words after the target word

CBoW

1. Considers the embeddings of 'h' words before and 'h' words after the target word
2. Adds them (order is lost) for predicting the target word

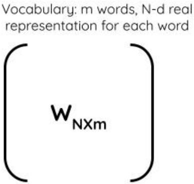


CBoW


The dog slept on couch
The
w(t-2)
dog
w(t-1)
SUM
w(t)
slept
on
w(t+1)
couch
w(t+2)

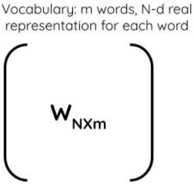
CBow

1. Size of the vocabulary $= m$



CBow

1. Size of the vocabulary $= m$
2. Dimension of the embeddings $= N$

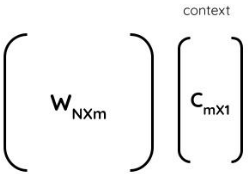


Word Embeddings: CBoW

1. Input layer $W_{m \times V}$ projects the context in to N-d space

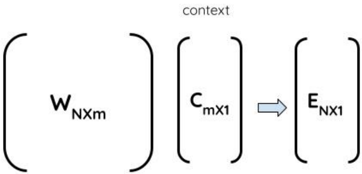
Word Embeddings: CBoW

1. Input layer $W_{m \times V}$ projects the context in to N-d space
2. Representations of all the $(2h)$ words in the context are summed (result is an V-d context vector)



Word Embeddings: CBoW

1. Input layer $W_{N \times m}$ projects the context in to N-d space
2. Representations of all the $(2h)$ words in the context are summed (context is an V-d vector)

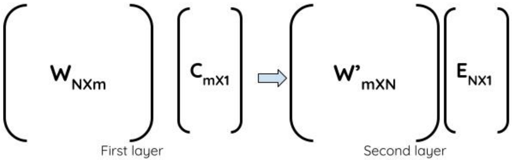


Word Embeddings: CBoW

1. Next layer has a weight matrix $W'_{m \times N}$

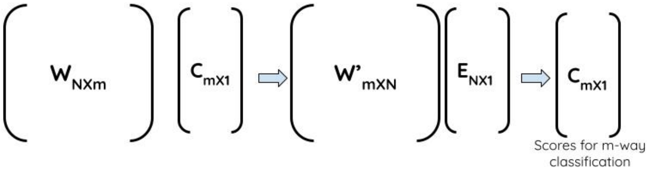
Word Embeddings: CBoW

1. Next layer has a weight matrix $W'_{m \times N}$
2. Projects the accumulated embeddings onto the vocabulary



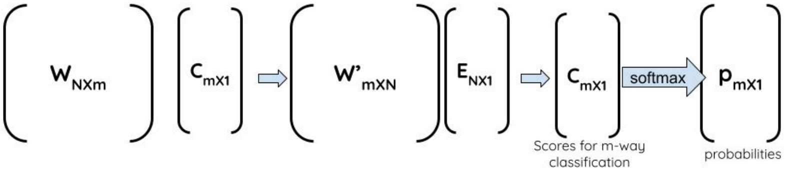
Word Embeddings: CBoW

1. Next layer has a weight matrix $W'_{V \times N}$
2. Projects the accumulated embeddings onto the vocabulary



Word Embeddings: CBoW

1. V- way classification $\rightarrow$ (after a softmax) maximizes the probability for the target word



Word Embeddings: CBoW

1. $W_{N \times m}$ is the W_{context}

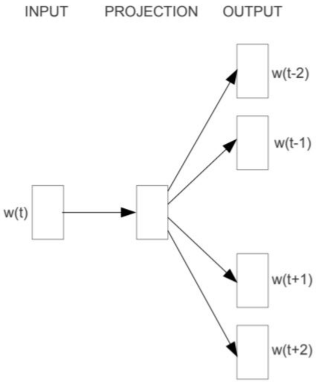
Word Embeddings: CBoW

1. $W_{N \times m}$ is the W_{context}
2. $W'_{m \times N}$ is the W_{words}

CBoW: issues

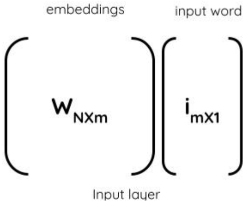
1. Softmax at the o/p is very expensive $\hat{y}_w = \frac{exp(u_c \cdot v_w)}{\sum_{w' \in V} exp(u_c \cdot v_{w'})}$

Word Embeddings: Skip-gram

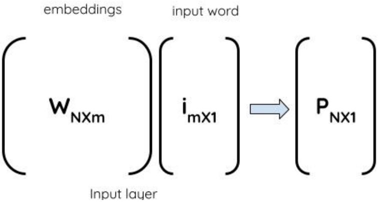


Skip-gram

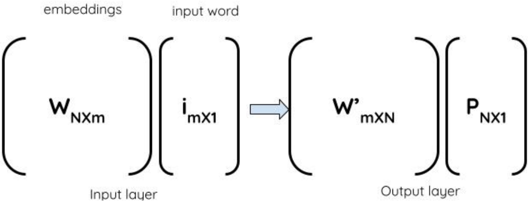
Word Embeddings: Skip-gram



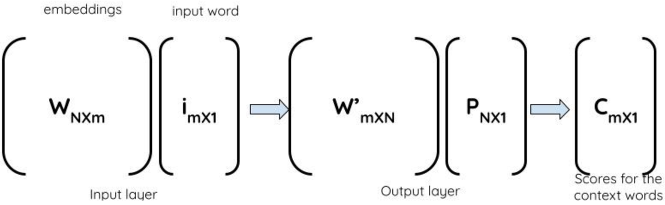
Word Embeddings: Skip-gram



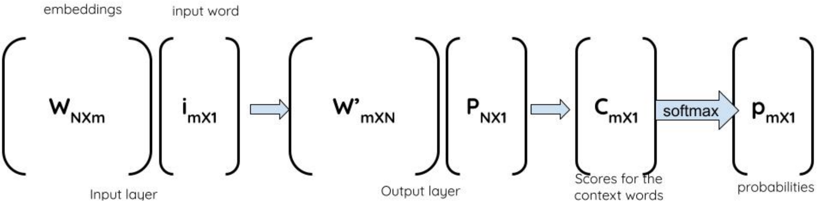
Word Embeddings: Skip-gram



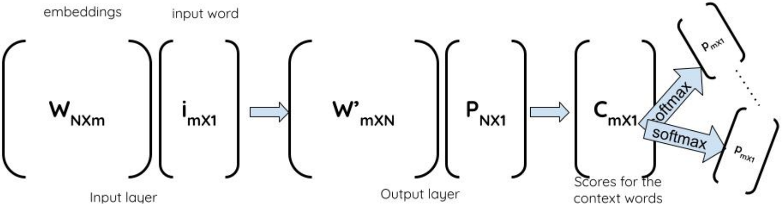
Word Embeddings: Skip-gram



Word Embeddings: Skip-gram



Word Embeddings: Skip-gram



Word Embeddings: Skip-gram

1. Loss is the summatoin of $\mathbb{L}(\theta) = -\sum_{i=1}^{2h} \log(\hat{y}_{w_i})$

Word Embeddings: Skip-gram

1. Loss is the summatoin of $\mathbb{L}(\theta) = -\sum_{i=1}^{2h} \log(\hat{y}_{w_i})$
2. $W_{N\times m}$ is the W_{word}

Word Embeddings: Skip-gram

1. Loss is the summatoin of $\mathbb{L}(\theta) = -\sum_{i=1}^{2h} \log(\hat{y}_{w_i})$
2. $W_{N \times m}$ is the W_{word}
3. $W'_{m \times N}$ is the W_{context}

Skip-gram: Issues

1. Expensive softmax operation at the o/p (same as that of CBoW)

Skip-gram: Issues

1. Expensive softmax operation at the o/p (same as that of CBoW)
2. Negative sampling: subset of incorrect words participate (instead of all)

Skip-gram: Issues

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

1. Expensive softmax operation at the o/p (same as that of CBoW)
2. Negative sampling: subset of incorrect words participate (instead of all)
3. Other solutions: Contrastive estimation, and hierarchical softmax

Glove

1. Glove - Global Vectors

Glove

1. Glove - Global Vectors
2. Combines the score based and predict based approaches

Glove

1. X_{ij} in the cooccurrence matrix encodes the global info. about words i and j

Glove

1. X_{ij} in the cooccurrence matrix encodes the global info. about words i and j
2. Glove attempts to learn representations that are faithful to the cooccurrence info.

Glove

1. X_{ij} in the cooccurrence matrix encodes the global info. about words i and j
2. Glove attempts to learn representations that are faithful to the cooccurrence info.
3. $v_i^T v_j = \log P(j/i) = \log X_{ij} - \log X_i$

Glove

1. X_{ij} in the cooccurrence matrix encodes the global info. about words i and j
2. Glove attempts to learn representations that are faithful to the cooccurrence info.
3. $v_i^T v_j = \log P(j/i) = \log X_{ij} - \log X_i$
4. $v_j^T v_i = \log P(i/j) = \log X_{ij} - \log X_j$

Glove

1. (add the two equations) $v_i^T v_j = \log X_{ij} - \frac{1}{2}\log X_i - \frac{1}{2}\log X_j$

Glove

1. (add the two equations) $v_i^T v_j = \log X_{ij} - \frac{1}{2}\log X_i - \frac{1}{2}\log X_j$
2. Since $\log X_i$ and $\log X_j$ depend on the words i and j, they can be considered as the word specific biases

Glove

1. (add the two equations) $v_i^T v_j = \log X_{ij} - \frac{1}{2}\log X_i - \frac{1}{2}\log X_j$
2. Since $\log X_i$ and $\log X_j$ depend on the words i and j, they can be considered as the word specific biases
3. $v_i^T v_j + b_i + b_j = \log X_{ij}$

Glove

1. (add the two equations) $v_i^T v_j = \log X_{ij} - \frac{1}{2}\log X_i - \frac{1}{2}\log X_j$
2. Since $\log X_i$ and $\log X_j$ depend on the words i and j, they can be considered as the word specific biases
3. $v_i^T v_j + b_i + b_j = \log X_{ij}$
4. $\sum_{i,j}\left(v_i^T v_j + b_i + b_j - \log X_{ij}\right)^2$

Evaluating the embeddings

1. Semantic relatedness (compute the correlation with humans' similarity)

Evaluating the embeddings

1. Semantic relatedness (compute the correlation with humans' similarity)
2. Synonym detection

Evaluating the embeddings

1. Semantic relatedness (compute the correlation with humans' similarity)
2. Synonym detection
3. Analogy

Comparison among different models

1. Difficult to judge!

Comparison among different models

1. Difficult to judge!
2. Some studies favor the predict-based, some the cooccurrence based!!