Representing Words with Vectors

Giri Iyengar

Cornell University

gi43@cornell.edu

Feb 21, 2018

Agenda for the day

- Learning Word Representations
- GloVe model
- Skip-Grams
- CBOW
- FastText

Overview

Word Representations

- Number of words in human language are far too numerous
- One-hot encoding doesn't capture relationships between words
- Compact representations would make the math work easier / training models easier
- Would be useful to capture Synonyms / Homonyms / Antonyms in these representations
- Would be useful to capture other relationships (e.g. King:Queen :: Man:Woman)

Word Representations: Some Assumptions

- Words that appear in similar contexts have similar meaning
- Co-occurrence of words convey meaning / structure of language
- Sub-word structures exist in languages

Word Representations

Goal

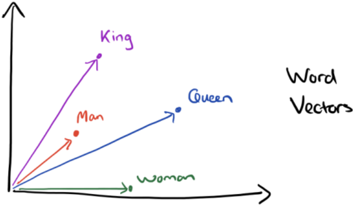
If we convert words into vectors in such a way that words with similar meanings will have vectors that lie nearby; Further if we can do vector arithmetic on them, it would be great. E.g. King - Man + Woman = Queen

Word Representations

Representing Words by Vectors

We want to do something like: King $\rightarrow (0.3, 0.9, 0.9, 0.2)$, Queen $\rightarrow (0.3, 0.9, 0.1, 0.2)$ etc

Word Representations



Overview

GloVe: Global Vectors for Word Representations

Pr. and Ratio	k=solid	k=gas	k=ice	k=steam
$P(k\|ice)$	$1.9E-4$	$6.6E-5$	$3.0E-3$	$1.7E-5$
$P(k\|steam)$	$2.2E-5$	$7.8E-4$	$2.2E-3$	$1.8E-5$
$P(k\|ice)/P(k\|steam)$	8.9	$8.5E-2$	1.36	0.96

Table: Co-occurrence Probabilities and their ratios from a 6 Billion word corpus

GloVe model motivation

- Perhaps model a pair of words and their context as $F(w_i, w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}$, where w is the vector representation we desire

GloVe model motivation

- Perhaps model a pair of words and their context as $F(w_i, w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}$, where w is the vector representation we desire
- Futher, we want F to encode vector arithmetic, suggesting $F(w_i - w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}$

GloVe model motivation

- Perhaps model a pair of words and their context as $F(w_i, w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}$, where w is the vector representation we desire
- Futher, we want F to encode vector arithmetic, suggesting $F(w_i - w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}$
- We also want to keep things linear, if possible: $F((w_i - w_j)^T \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}$

GloVe model motivation

- Perhaps model a pair of words and their context as $F(w_i, w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}$, where w is the vector representation we desire
- Futher, we want F to encode vector arithmetic, suggesting $F(w_i - w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}$
- We also want to keep things linear, if possible: $F((w_i - w_j)^T \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}$
- The distinction between w and $\tilde{w}$ is arbitrary. Applying Homomorphism: $F((w_i - w_j)^T \tilde{w}_k) = \frac{F(w_i^T \tilde{w_k})}{F(w_j^T \tilde{w_k})}$

GloVe model motivation

- This suggests: $F(w_i^T \tilde{w}_k) = P_{ik}$

GloVe model motivation

- This suggests: $F(w_i^T \tilde{w}_k) = P_{ik}$
- One solution: $w_i^T \tilde{w}_k = \log(P_{ik}) = \log(X_{ik}) - \log(X_i)$, where X_{ik} is the co-occurrence count

GloVe model motivation

- This suggests: $F(w_i^T \tilde{w}_k) = P_{ik}$
- One solution: $w_i^T \tilde{w}_k = \log(P_{ik}) = \log(X_{ik}) - \log(X_i)$, where X_{ik} is the co-occurrence count
- You can make this a *completely* symmetric model by introducing appropriate bias: $w_i^T \tilde{w}_k + b_i + \tilde{b}_k = \log(X_{ik})$

GloVe model motivation

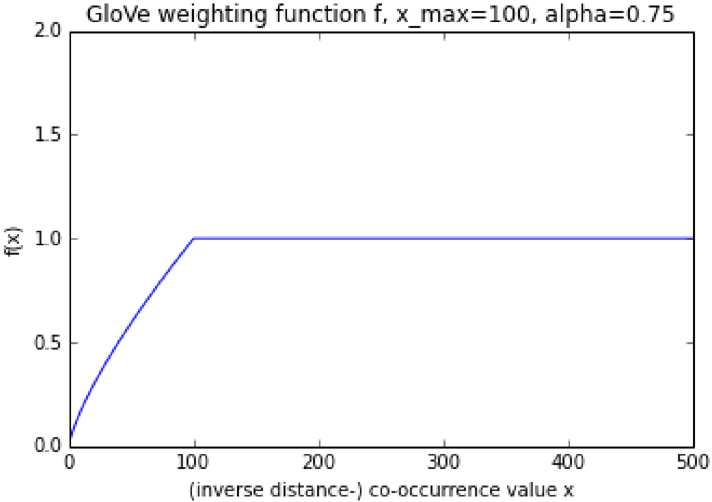
- This suggests: $F(w_i^T \tilde{w}_k) = P_{ik}$
- One solution: $w_i^T \tilde{w}_k = \log(P_{ik}) = \log(X_{ik}) - \log(X_i)$, where X_{ik} is the co-occurrence count
- You can make this a *completely* symmetric model by introducing appropriate bias: $w_i^T \tilde{w}_k + b_i + \tilde{b}_k = \log(X_{ik})$
- When $X_{ik} = 0$, this is ill-defined.

GloVe model motivation

- This suggests: $F(w_i^T \tilde{w}_k) = P_{ik}$
- One solution: $w_i^T \tilde{w}_k = \log(P_{ik}) = \log(X_{ik}) - \log(X_i)$, where X_{ik} is the co-occurrence count
- You can make this a *completely* symmetric model by introducing appropriate bias: $w_i^T \tilde{w}_k + b_i + \tilde{b}_k = \log(X_{ik})$
- When $X_{ik} = 0$, this is ill-defined.
- Introduce a weighting function $f(X_{ik})$, giving us a new loss function to minimize:

GloVe model motivation

- This suggests: $F(w_i^T \tilde{w}_k) = P_{ik}$
- One solution: $w_i^T \tilde{w}_k = \log(P_{ik}) = \log(X_{ik}) - \log(X_i)$, where X_{ik} is the co-occurrence count
- You can make this a *completely* symmetric model by introducing appropriate bias: $w_i^T \tilde{w}_k + b_i + \tilde{b}_k = \log(X_{ik})$
- When $X_{ik} = 0$, this is ill-defined.
- Introduce a weighting function $f(X_{ik})$, giving us a new loss function to minimize:
- $J = \sum_{i,k}^{V} f(X_{ik})(w_i^T \tilde{w}_k + b_i + \tilde{b}_k - \log(X_{ik}))^2$

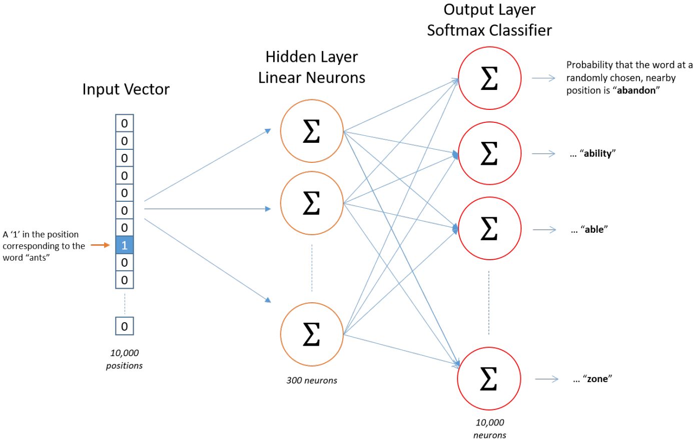

GloVe weighting function f, x_max=100, alpha=0.75
2.0
1.5
1.0
0.5
0.0
f(x)
0
100
200
300
400
500
(inverse distance-) co-occurrence value x

Overview

Skip-gram Model

Predict a context word, given an input word

Given *brown*, what is the probability that *the*, *quick*, *fox*, *jumps* appear in its neighborhood in a sentence


Input Vector
0
0
0
0
0
0
0
1
0
0
0
10,000 positions
A '1' in the position corresponding to the word "ants"
Hidden Layer
Linear Neurons
300 neurons
Output Layer
Softmax Classifier
10,000 neurons
Probability that the word at a randomly chosen, nearby position is "abandon"
... "ability"
... "able"
... "zone"

Source Text	Training Samples
The quick brown fox jumps over the lazy dog.	(the, quick) (the, brown)
The **quick** brown fox jumps over the lazy dog.	(quick, the) (quick, brown) (quick, fox)
The quick **brown** fox jumps over the lazy dog.	(brown, the) (brown, quick) (brown, fox) (brown, jumps)
The quick brown **fox** jumps over the lazy dog.	(fox, quick) (fox, brown) (fox, jumps) (fox, over)

Skip-gram Model Training

- $P(w_c|w_t) = \frac{\exp^{s(w_t, w_c)}}{\sum_{j=1}^{V} \exp^{s(w_t, w_j)}}$
- Online training, using SGD

Skip-gram Model Training

- Consider special n-grams as single words: New York, Boston Globe
- Negative sampling to selectively at random update a few negative samples. Frequent words have a higher chance of being selected for negative sampling
- Sub-sample frequent words

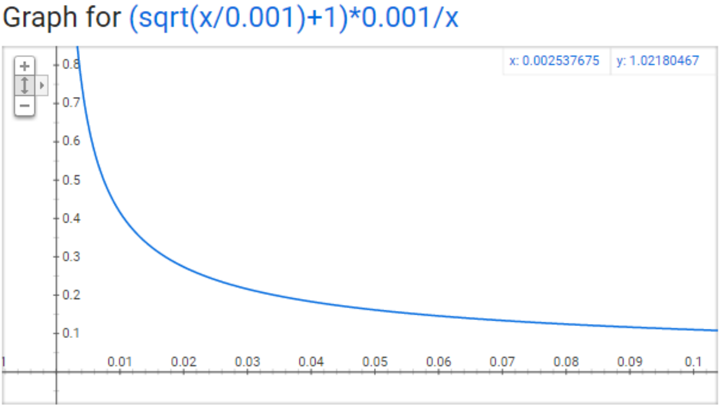
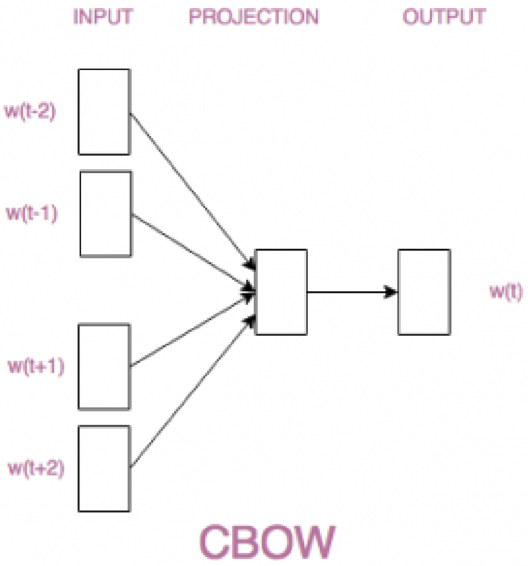


Figure: Plot of word frequency and Probability of keeping. Empirically obtained

Overview

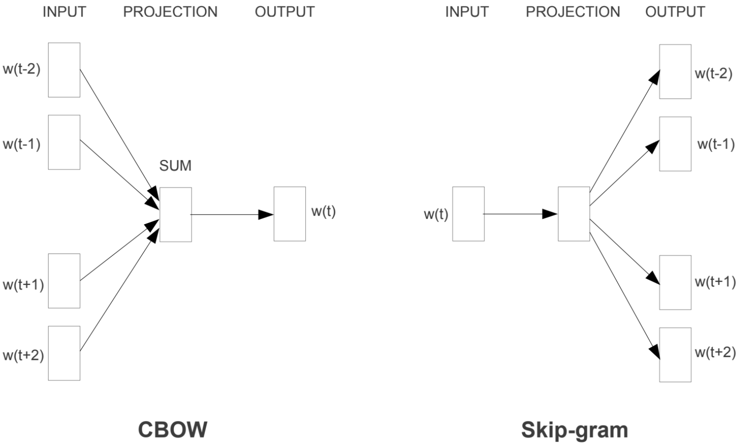
CBOW Model

- Instead of predicting the *context*, predict the *target* given the context
- Given (the, quick, fox, jumps), predict $\widehat{\text{brown}}$


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

CBOW Model

- Works better than Skip-gram when corpus is smaller
- Embeddings are averaged across the context, perhaps resulting in more stable representations


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

Overview

FastText Model

- All models presented so far, model whole words

FastText Model

- All models presented so far, model whole words
- They all ignore sub-word structures

FastText Model

- All models presented so far, model whole words
- They all ignore sub-word structures
- Many languages have distinct structures for words

FastText Model

- All models presented so far, model whole words
- They all ignore sub-word structures
- Many languages have distinct structures for words
- Many word forms occur rarely even in large corpora, preventing learning good representations for them

FastText Model

- Use character n-grams

FastText Model

- Use character n-grams
- E.g. *Where* is modeled as (<wh,whe,her,ere,re>,<where>)

FastText Model

- Use character n-grams
- E.g. *Where* is modeled as (<wh,whe,her,ere,re>,<where>)
- $s(w, c) = \sum_{g \in G_w} z_g^T v_c$

FastText Model

- Use character n-grams
- E.g. *Where* is modeled as (<wh,whe,her,ere,re>,<where>)
- $s(w,c) = \sum_{g \in G_w} z_g^T v_c$
- $P(c|w) = \frac{\exp^{s(w,c)}}{\sum_{j=1}^{V} \exp^{s(w,j)}}$


thank you
danke
謝謝
ngiyabonga
teşekkür ederim
gracias
спасибо
рахмат
Баярлалаа
bedankt
dziękuję
obrigado
hvala
mauruuru
dank je
tapadh leat
хвала
asante
manana
mochchakkeram
go raibh maith agat
sukriya
kop khun krap
terima kasih
grazie
arigatô
takk
dakujem
мерси
merci
감사합니다
ありがとう
ευχαριστώ
sagolun
taiku
enkosi