Learning (Training) Word Embedding

method 1: Bengio et. al., 2003, A neural probabilistic language model

I	want	a	glass	of	orange	——. (total words: 10,000)
4343	9665	1	3852	6163	6257	

I $o_{4343} \to E \to e_{4343}$

(denoted a one-hot)

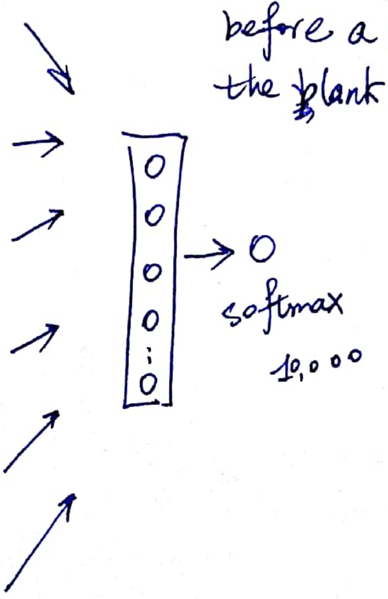
want $o_{9665} \to E \to e_{9665}$

a $o_1 \to E \to e_1$

glass $o_{3852} \to E \to e_{3852}$

of $o_{6163} \to E \to e_{6163}$

orange $o_{6257} \to E \to e_{6257}$



Given a few words before a blank, predict the blank.

method 2: Mikolov et. al., 2013 Efficient estimation of word representation in vector space (Word2Vec)

I want a glass of $\underset{}{\overset{\text{context}}{\underline{\text{orange}}}}$ juice to go along with my cereal

skip grams: randomly choose target words within a given window size.

context	target
orange	juice
orange	glass
orange	my
⋮	⋮

supervised learning problem

context → target

How to sample context?
need to find a balance between common words and less common words.

Learning (Training) Word Embedding - 2

Total vocab size = 10,000

$O_{context} \rightarrow E \rightarrow O_{context} \rightarrow O_{softmax} \rightarrow \hat{y}$

Softmax:

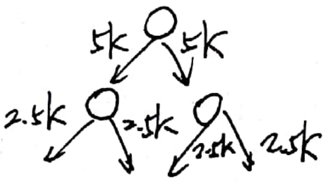
$$p(t|c) = \frac{e^{\theta_t^T e_c}}{\sum_{j=1}^{10,000} e^{\theta_j^T e_c}}$$

👎 problems with softmax classification

$\sum_{j=1} e^{\theta_j^T e_c} \leftarrow O(n)$ n: number of vocabs

↓

Hierarchial Softmax



$\log^{|n|}$

method 3: Mikolov et. al., 2013. Distributed representation of words and phrases and their compositionality

(Negative Sampling)

I want a glass of orange juice to go alone with my cereal.

context word: a word in the sentence

positive word pair: pair the context word with a random chosen word within the window size

negative word pair: pair the context word with a random chosen word from the dictionary

ex:

pair	label
orange, juice	1
orange, king	0
orange, book	0

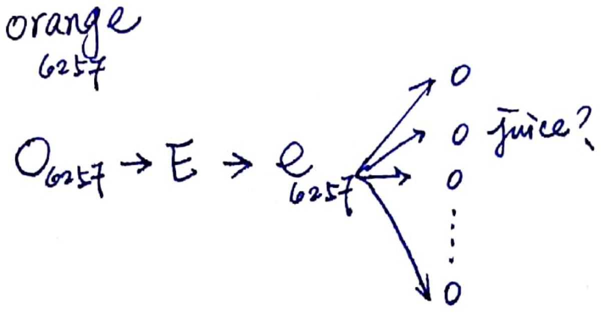
pick 1 context word
generate 1 positive pair
& generate k negative pairs

k: 5-20 smaller dataset
k: 2-5 larger dataset

Learning Training Word Embedding -3

Model (# vocab = 10,000)

x context	word	y target
orange	juice	1
orange	king	0
orange	book	0
orange	the	0
orange	of	0



10,000 binary logistic regression problems
but each iteration only train (k+1) of them

$$P(y=1 \mid c, t) = \sigma(\theta_t^T e_c)$$

How to sample negative pairs?

$$p(w_i) \longleftrightarrow \frac{f(w_i)^{3/4}}{\sum_{j=1}^{10000} f(w_j)^{3/4}} \longleftrightarrow \frac{1}{|V|}$$

sample
according to the empirical frequency
(E.g. a. the. of ... tend to be picked)

uniformly
sample
(E.g. not representative)

Method 4 : GloVe (global vectors for word representation)

(Pennington et. al. 2014 GloVe : Global vectors for word representation)

I want a glass of orange juice to go along with my cereal.

X_{ij} = # times $\underline{i}$ appears in context of $\underline{j}$

i: target word

j: context word

(How often word i and word j appear together)

Model

$$\text{minimize} \sum_{i=1}^{10000} \sum_{j=1}^{10000} \underline{f(X_{ij})} \left(\underline{\theta_i^T e_j} + b_i + b_j' - \log X_{ij} \right)^2$$

→ θ_i e_i are symmetric

→ $e_w^{(final)} = \frac{e_w + \theta_w}{2}$

a weighting term

- $f(X_{ij}) = 0$ if $X_{ij} = 0$
- used to find a balance between common words and non-common ones

Scanned by CamScanner