

(Week 2) (Word Representations)

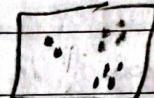
→ Word representations

- word embedding is a way of representing words

- instead of one-hot representation we instead learn a featureized representation : word embedding

↳ if we use it to represent the word apple & orange we will notice that they are quite similar & this increases the odds of the learning alg. that has figured out that orange juice and apple juice are different things

- if we able to learn 300 Dim. embedding for each word, we take the 300D & embed it in 2D dim space so we can visualize them using : t-SNE alg.



→ Using word Embeddings

- How we take featureized representation & plug them into NLP applications?

→ properties of word embeddings

- Dense representation, Semantic Similarity, Contextual relation

→ Embedding Matrix

- when implement an algorithm to learn a word embedding it end up learning is an embedding matrix

- stores vector representation for all the words in the vocab.

each row in the matrix is correspond to a word

→ Learning word embeddings.

.. will talk about some concrete algorithms for learning word embeddings

→ is way to learn a set of embeddings

Natural language model to predict a word

I want a glass of orange

4343 9665 1 385 6152 6257

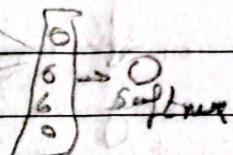
I want
a
glass
of
orange

ohv Matrix Param E

embedding matrix

$$E = \begin{pmatrix} e_{1343} \\ e_1 \\ e_2 \\ e_3 \\ e_4 \end{pmatrix}$$

embedding vector



→ Word2Vec model (ex: skipgrams)

it is more simple & efficient way to learn embeddings

ex: I want a glass of orange juice to go along --

in the Skipgram model, what we're going to do is come up with a few [context] $\xrightarrow{\text{to}}$ [target] to create supervised prob.

orange

Juice

$$x \rightarrow y$$

$$x \rightarrow t \rightarrow e_c \rightarrow \hat{y} \xrightarrow{\text{softmax}}$$

target is chosen randomly within some window

$l(\hat{y} - y)$ if we optimize the loss func. with respect to

all of these parameters we get good embedding vectors

it is called Skipgram as it input one word (orange) & predict some words & skipping few words the left or right side & problems

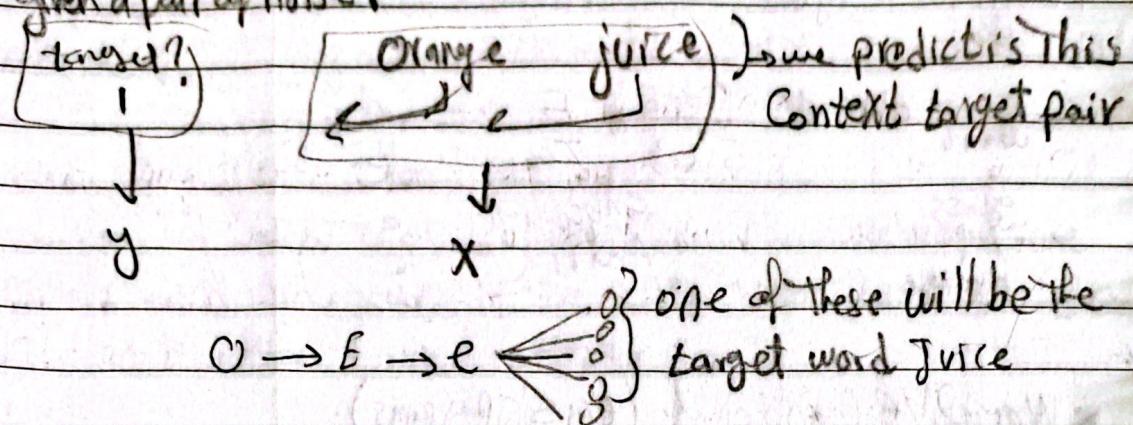
of Skipgram is Softmax classification as it is computationally expensive

$$P(t/c) = \frac{e^{o_t c}}{\sum_{j=1}^n e^{o_j c}}$$

→ Negative Sampling (technique used to optimize the training of word2vec models)

- Will talk about modified learning problem called negative sampling that allows you to do something similar to skipgram model but with much more efficient learning algorithm.

- Given a pair of noise:



instead of having 10,000 way softmax, we've instead turned it into 10,000 binary classification problems.

Context word	target?	it is called
orange	juice	neg. example
orange	king	bec. what you're doing
orange	book	is you have the example
orange	the	& generate bunch of neg. examples

How do you choose the (-ve) examples
 ① by how often different words appears but the problem with that it ends up with very high representation of words
 ② $\frac{1}{|V|} \rightarrow$ vocab size

(The meaning of sampling 5)

→ Glove word vectors.

Sentiment Classification

- It is a task of looking at a piece of text & telling if someone likes or dislikes the thing they're talking about
 - One of the challenges of sentiment classification is might not having huge labeled training dataset for it, but with word embeddings you're able to build good sentiment classifiers even with only modest-size label training sets

Debiasing word embeddings

- is the process of reducing or removing bias associations enc
in the word embeddings summarized as stereotypes
 - we don't mean the bias variants, instead gender, ethnicity
sexual orientation bias.