# Doubts of CS224n Lecture - 1

Q) Asked by Vignesh

Refined question: Does it mean that the vectors which have large magnitude are more likely to be selected as the context word, or in other words P(o|c) for it will be larger? If so, why?

Answer: Your intuition is correct. Yes, the words with larger magnitudes will more likely be associated as context word. Mathematically, we can see that, too; the numerator will be more significant for those words. Note that the normalization (denominator) is the same for all the words of the vocabulary given a centre word c.

The answer lies in the fact that there is also a meaning associated with the length of the word vectors,  similar to what is associated with directions (eg boy and girl are expected to be in opposite directions).

When a word appears in different contexts, its vector gets moved in different directions during updates. The final vector then represents some sort of weighted average over the various contexts. Averaging over vectors that point in different directions typically results in a vector that gets shorter with the increasing number of different contexts in which the word appears. For words to be used in many different contexts, they must carry little meaning. Prime examples of such insignificant words are high-frequency stop words, which are indeed represented by short vectors despite their high term frequencies.

For a given term frequency, the vector length is seen to take values only in a

narrow interval. That interval initially shifts upwards with increasing frequency. Around a frequency of about 30, that trend reverses, and the interval shifts downwards
Both forces determining the length of a word vector are seen at work here. Small-frequency words tend to be used consistently, so that the more frequently such words appear, the longer their vectors. High-frequency words, on the other hand, tend to be used in many different contexts, the more so, the more frequently they occur. The averaging over an increasing number of different contexts shortens the vectors representing such words.

Well, this also highlights one of the major disadvantages of word2vec. There is only one vector for all the different meanings a word might have. This problem is known as word sense disambiguation. For e.g.: The word 'bank' can be associated with money bank or river bank. But word2vec has only one vector for this word. Also, it depends on the text on which word2vec is trained, which meaning it will convey more strongly. Like, in this case, if the text corpus used for training word2vec is financial magazines, then you might expect 'bank' to be more influenced with money bank. Keep in mind this problem while continuing with the course!!

Q) How the heck were the higher dimensional word vectors were visualized in 2d?

Answer: It is called dimensionality reduction. Google up this term and read up on PCA. The basic idea is that, out of 100s of dimensions, the algorithm chooses two axes which carry the highest variability. Then the entire data is projected onto these two dimensions. This generally results in a decrease in information, but nevertheless, it is super-effective for visualization and

tackling the curse of dimensionality.

Q) What is theta in the equations they were using?

Answer) In machine learning, we try to estimate the parameters which might represent the underlying distribution of data, using the data we are given. If you recall your PnS classes, there used to be terms like sample mean. To estimate the mean of the whole event, you take some samples, calculate its mean and assume that the underlying sample space will have approximately the same mean. That is called parameter in machine learning, commonly represented with theta.

Specifically, in the word2vec, the parameter is nothing but a 2DV dimensional data.

Significance of each numeral:

2 : Remember, we used two vectors for each word. We call it v when it is the centre word, and we call it u when it is a context word. This is done just for computational simplicity. Hence for each word, there are two vectors.

d: d is the dimensionality of each vector (e.g. 100). Remember, we were trying to learn a dense vector for each word which could represent its meaning, starting from its one-hot encoding. So, for example, if we had a vocabulary of 10,000 words, then every word is represented initially as 10,000 X 1 vector, which has one 1 and rest all 0s. After training the representation of that word is 100 X 1, where all values are now real values (not necessarily 0 and 1).

V : the number of words in the vocabulary.

Hence the theta vector looks something like:

Vaabra
Vbaseball
.
.
.
.
.
Vzebra
Uaabra
Ubaseball
.
.
.
.
.
Uzebra

Each of the entries, Vaabra, is a vector of d dimensions. For each of u and v, there are V words. Hence there are V rows for each of u and v. Hence the size becomes 2dV.

I hope this clears things a bit. Feel free to shoot in any follow-up questions

to me. I will be happy to help :)