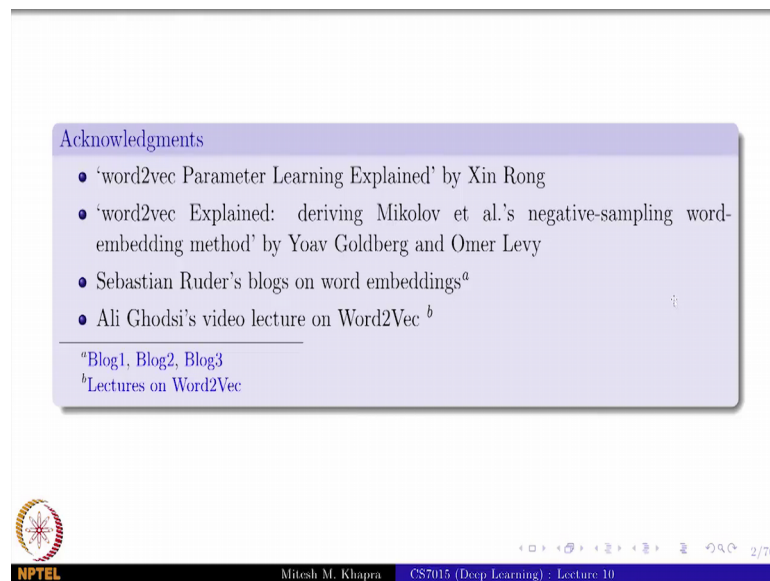


Deep Learning
Prof. Mitesh M Khapra
Department of Computer Science Engineering
Indian Institute of Technology, Madras

Lecture – 10
Learning Vectorial Representations of Words

So, today we are going to talk about learning vectorial representations for words.

(Refer Slide Time: 00:18)



A slide titled "Acknowledgments" with a light purple background. It contains a bulleted list of references:

- 'word2vec Parameter Learning Explained' by Xin Rong
- 'word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method' by Yoav Goldberg and Omer Levy
- Sebastian Ruder's blogs on word embeddings^a
- Ali Ghodsi's video lecture on Word2Vec^b

Footnotes:

^aBlog1, Blog2, Blog3
^bLectures on Word2Vec

The slide also features the NPTEL logo in the bottom left corner, a navigation bar with icons, and the text "Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 10" in the bottom right corner.

So, these are the acknowledgement slash references for where are the things that I have referred to by preparing for this lecture. So, you can just go this some of these are also available as video lectures on YouTube can take a look at them also.

Say in the first module, we are going to look at one hot representations of words ok.

(Refer Slide Time: 00:39)

• Let us start with a very simple motivation for why we are interested in vectorial representations of words

• Suppose we are given an input stream of words (sentence, document, etc.) and we are interested in learning some function of it (say, $\hat{y} = \text{sentiments}(\text{words})$)

\mathbb{R}^n $\{0,1\}^n$

This is by far AAMIR KHAN's best one. Finest casting and terrific acting by all.

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 10

So, as usual we will start with this motivation or motivation, motivating question why do we need to learn representations for words. Vectorial representations for words, and vectorial representations for words when words are there right, you can write them using alphabets and characters and so on. So, why do we need vectorial representations? Mention whatever you have seen so far in the course, I have seen anything like this let us see.

So, suppose you are given an input stream of words, and it could be a sentence or documented. If I say documented pretty much covers almost all the text that you see right, you can always abstract everything as a document and email is also a document, a manuals are also documents and so on.

And we are interested in learning some function of it. And so, I am given a document and I am interested in the function \hat{y} . Say \hat{y} is equal to sentiments of the words in the document or the sentiment of the document itself.

This is imaginable this is not something that I am cooking up this is something that, you would want to do you would log on to for example, if you are a movie maker you would want to know once the movie is released people have written reviews about it what is a sentiment is positive or negative. Similarly, if Apple has released a new product or a new feature you would want to know what are the reviews written about this product and what is the feature what is the sentiment coming out of it is positive or negative, right?

Now, sentiment is a binary thing, or it could be rated also right it could be on a scale of one to 10 also, but let us consider it is binary that either people liked it or did not like it.

So now I am trying to learn this function, which gives me which takes as input words, but as output gives me real numbers, either 0 to 1 or on a scale or 1 to 10 or whatever. And this is not something that we have dealt within the course so far, let how do we take as input word. So, all inputs have already always been numbers, right? They were either coming from \mathbb{R}^n or they are coming from 0 to 1 raise to n or something of that sort.

We never had the situation when we have words as written. So, right so now how do we deal with the situation and also I have made a case for that learning this function is a valid thing to do, you have several news cases where you will need this

(Refer Slide Time: 02:41)

- Let us start with a very simple motivation for why we are interested in vectorial representations of words
- Suppose we are given an input stream of words (sentence, document, etc.) and we are interested in learning some function of it (say, $\hat{y} = \text{sentiments}(\text{words})$)
- Say, we employ a machine learning algorithm (some mathematical model) for learning such a function ($\hat{y} = f(\mathbf{x})$)
- We first need a way of converting the input stream (or each word in the stream) to a vector \mathbf{x} (a mathematical quantity)

So now, if we employ a machine learning algorithm, that some mathematical model; so we saw that we could have several such models logistic regression svm and neural network and feed forward neural networks and so on, right? And at the end we are trying to learn this function \hat{y} is equal to f of x , but in our case, the x instead of have be instead of x being numbers it turns out that x is actually a collection of words right.

So now how do we reconcile with the situation where we have suddenly have words instead of numbers. So, the way to do that would be we need a way of converting these words or documents into some number, into some vectorial representation and once we

have this vectorial representation right. So now, we have R , R raised to n and we know how to deal with R raised to n given R raised to n , how to predict R or even R square or R^m in general, that we know right. We can design neural networks or any other machine should we calculate, but how do we go from here to here that is the question, right? And that is why we need to learn vectorial representations of words.

This is a motivation clear to everyone? Okay. Now let us start getting with a (Refer Time: 03:52) how to do that right.

(Refer Slide Time: 03:54)

Corpus:

- Human machine interface for computer applications
- User opinion of computer system response time
- User interface management system
- System engineering for improved response time

$V =$ [human, machine, interface, for, computer, applications, user, opinion, of, system, response, time, interface, management, engineering, improved]

machine:

0	1	0	...	0	0	0
---	---	---	-----	---	---	---

- Given a corpus, consider the set V of all unique words across all input streams (*i.e.*, all sentences or documents)
- V is called the **vocabulary** of the corpus (*i.e.*, all sentences or documents)
- We need a representation for every word in V
- One very simple way of doing this is to use one-hot vectors of size $|V|$

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 10

So now we will start hearing this word corpus. Have you heard this word before? That is exactly what you are collecting for the word to like assignment right you are collecting a corpus in specific languages. And you have taken a very Toyish corpus for the purpose of illustration. So, here is a corpus it just contains 4 sentences. So, think of it that I have a very restricted domain, I have very small set of documents and I just have these 4 sentences with me, this is the valid corpus.

The corpus that you have constructing is probably much larger scale you are trying to collect 100 thousand sentences or 50 thousand sentences or something of that order right. Ah, But we will take this toy example.

Now, consider set V of all unique words across all these input streams. So, I just call them input streams by input streams I mean; sentences or documents or whatever right.

You could take it as any sequence of words. And these are set of all unique words across all this input sentences that you have.

So, can you tell me what V is here? What would can you tell me some elements of V of the set V ? Human machine interface and so on. So, that is why, in fact this is the entire set V which is written on the left hand side right. And V is called the vocabulary of the corpus so; that means, everything in the corpus comes from this vocabulary, all sentences are constructed by arranging words from this vocabulary.

Some of you might always I mean find this very trivial, but I am just going over the basics. So, that at least the terminology is clear to everyone. And what we want is a representation for every word in V . So, that is the title of the lecture learning vectorial representations of words. So, for every word in our vocabulary whatever corpus we are dealing with the vocabulary would change, and for every word there you want to learn a representation for that word. So, that is what our quest yesterday.

And now one very simple way of doing this is right, you tell me you want a vector and that is all you care about. Here is a vector I will give you one hot representations. So, if a total number of words in my vocabulary is V . I just construct a vector of size V ok. And I have assigned a number to every word in my vocabulary right. So, I will say human is equal to 0 machine is equal to 1 interface is equal to 2 and so on.

And if you ask me for a vectorial representation of that word, I will just say take this the or vector of size V , and switch on the corresponding bit and anything else would be 0. Hence 1, hot right at any given point of time only one of the elements in the vector would be on. So, that is a simple one hot representation as this is a very simple recipe to get a vectorial representation of words, and for every word in your vocabulary.


(Refer Slide Time: 06:31)

cat:	0	0	0	0	0	1	0
dog:	0	1	0	0	0	0	0
truck:	0	0	0	1	0	0	0

$euclid_dist(\text{cat}, \text{dog}) = \sqrt{2}$
 $euclid_dist(\text{dog}, \text{truck}) = \sqrt{2}$
 $cosine_sim(\text{cat}, \text{dog}) = 0$
 $cosine_sim(\text{dog}, \text{truck}) = 0$

Problems:

- V tends to be very large (for example, 50K for PTB, 13M for Google 1T corpus)
- These representations do not capture any notion of similarity
- Ideally, we would want the representations of cat and dog (both domestic animals) to be closer to each other than the representations of cat and truck
- However, with 1-hot representations, the Euclidean distance between **any two words** in the vocabulary is $\sqrt{2}$
- And the cosine similarity between **any two words** in the vocabulary is 0



Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 10 6/70

Now, what is the drawback? Of this V tends to be very larger right. So, for example, there is a standard corpus known as the Penn Treebank corpus, which is used in various NLP applications for various reasons. And that corpus has a vocabulary of 50 k.

Google of course, operates at its own scale. So, they have a word 1 T corpus which has 13 million words. So, this is like all the most of the web pages that they have drawn constructed a vocabulary for that.

So now I am talking about, for every word representing it by a vector of size 13 million, theory does not work right there is too much of storage required for that. And if you look at that information and it is. So, redundant that is all 0 except for that one bit which is on.

And the other important problem is that these representations do not capture any notion of similarity. Other 3 words that I have shown you which do you want to have similar representations? Cat and dog why, because both of them are domestic animals right? Both of them are mammals. So, there are some things that you would want at least, at the minimum that the similarity between a cat and dog is more than the similarity between cat and truck.

Or alternately the distance between cat and dog is less than the distance between cat and truck. So now, once I start talking about vectors, I can talk about similarities like cosine

similarities or I can start talking about Euclidian distance. So, once anything I convert it to a vector I can start asking these questions other 2 questions. Which I am asking are valid right what would you expect to be the Euclidian distance between cat and dog as compared to cat and truck.

Now, what happens with the one hot representations? Take any 2 words in your corpus, any 2 what will be the Euclidian distance? What will it be? Square root of 2 right. For all the words. Take any 2 words in your corpus what will that cosine similarity, mean 0 because all these vectors are orthogonal right. So, the cosine similarity is going to be 0, but this is; that means, these vectors are not really capturing any information about the essence of the word right.

So, remember always we are interested even like that has been our philosophy right from auto encodes. And so, right or even principle component on either, they are always interested in learning meaningful representations which capture something fundamental about the NTT that we are trying to represent right, but here something like that is clearly not happened. So, that is not acceptable.