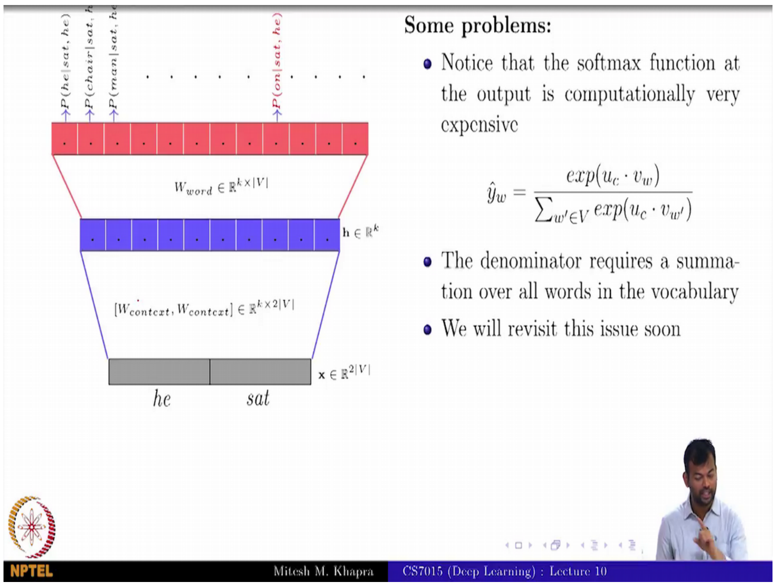
Deep Learning
Prof. Mitesh M Khapra
Department of Computer Science Engineering
Indian Institute of Technology, Madras

Module – 10.5
Lecture – 10
Skip-Gram Model

So, with that we will move on to the next model, I am still not telling you how to solve this problem; we will come to that later.

(Refer Slide Time: 00:15)



$P(he|sat,he)$
$P(chair|sat,he)$
$P(man|sat,he)$
$P(on|sat,he)$
$W_{word} \in \mathbb{R}^{k \times |V|}$
$h \in \mathbb{R}^k$
$[W_{context}, W_{context}] \in \mathbb{R}^{k \times 2|V|}$
$x \in \mathbb{R}^{2|V|}$
he
sat
NPTEL
Mitesh M. Khapra
CS7015 (Deep Learning) : Lecture 10

Some problems:

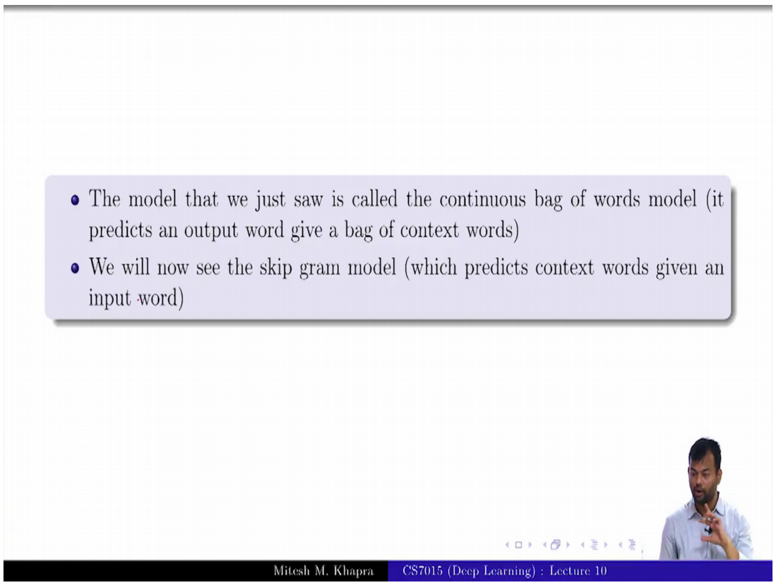
- Notice that the softmax function at the output is computationally very expensive

$$\hat{y}_w = \frac{exp(u_c \cdot v_w)}{\sum_{w' \in V} exp(u_c \cdot v_{w'})}$$

- The denominator requires a summation over all words in the vocabulary
- We will revisit this issue soon

I am just going to the next model which is the Skip gram model. And this is the famous word to make which you are trying to implement.

(Refer Slide Time: 00:23)

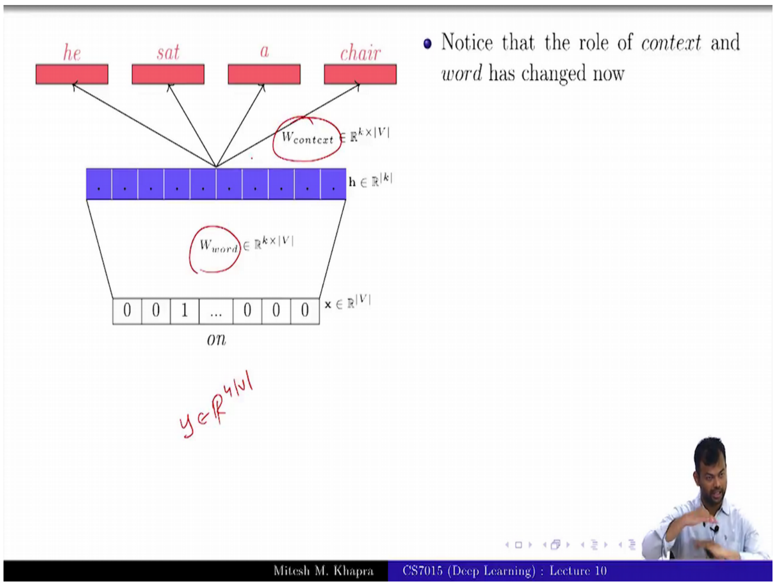


- The model that we just saw is called the continuous bag of words model (it predicts an output word give a bag of context words)
- We will now see the skip gram model (which predicts context words given an input word)

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

The model that we just saw was known as a continuous bag of words. It predicts the output given the context; skip gram model does the reverse of that.

(Refer Slide Time: 00:34)



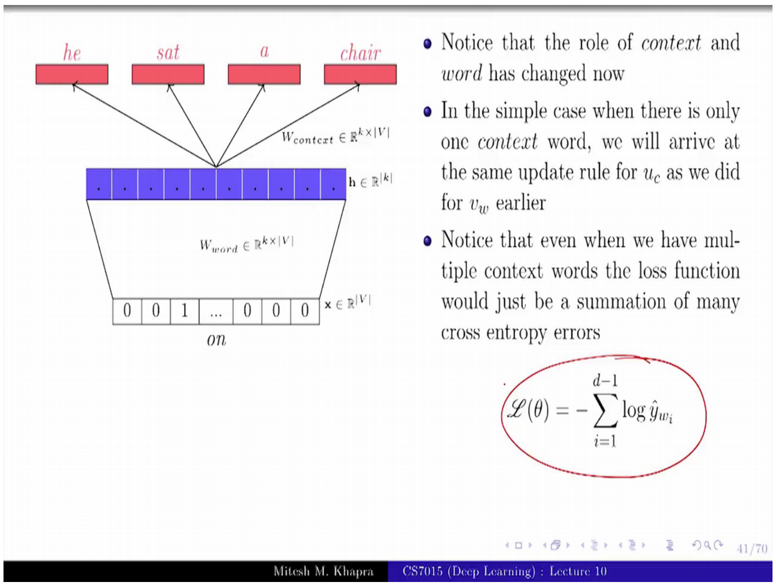
he
sat
a
chair
$W_{context} \in \mathbb{R}^{k \times |V|}$
$\mathbf{h} \in \mathbb{R}^{|k|}$
$W_{word} \in \mathbb{R}^{k \times |V|}$
0 0 1 ... 0 0 0
$\mathbf{x} \in \mathbb{R}^{|V|}$
on
$y \in \mathbb{R}^{4|V|}$
Mitesh M. Khapra
CS7015 (Deep Learning) : Lecture 10

- Notice that the role of *context* and *word* has changed now

You are given a word; you want to predict all the context words. So now, I am given the word on, I am trying to predict the words which appear on the left and right side of it is that fine. So, how many prediction problems am I solving? How many predictions am I giving you? 4 in this case right. So, you see that this is a case where your y actually belongs to R 4 right. Of course it is not R 4 it is 4 into V, and because you are predicting the entire distribution, but what I meant is that you want these 4 different outputs you just do not want one single output

Apart from that does everything else remain same? You have an input word you compute a hidden representation from that hidden representation you try to predict the outputs. You get a probability distribution, what is your loss function? It is a dash of cross entropies sum of cross entropies, how many cross entropies do you have? 4 in this case and also notice that I have, I hope I have yeah, I have changed W word and W context they are flipped now is that fine. The role of context and the word has changed.

(Refer Slide Time: 01:46)


he
sat
a
chair
$W_{context} \in \mathbb{R}^{k \times |V|}$
$h \in \mathbb{R}^{|k|}$
$W_{word} \in \mathbb{R}^{k \times |V|}$
0 0 1 ... 0 0 0
$x \in \mathbb{R}^{|V|}$
on
Mitesh M. Khapra
CS7015 (Deep Learning) : Lecture 10
41/70

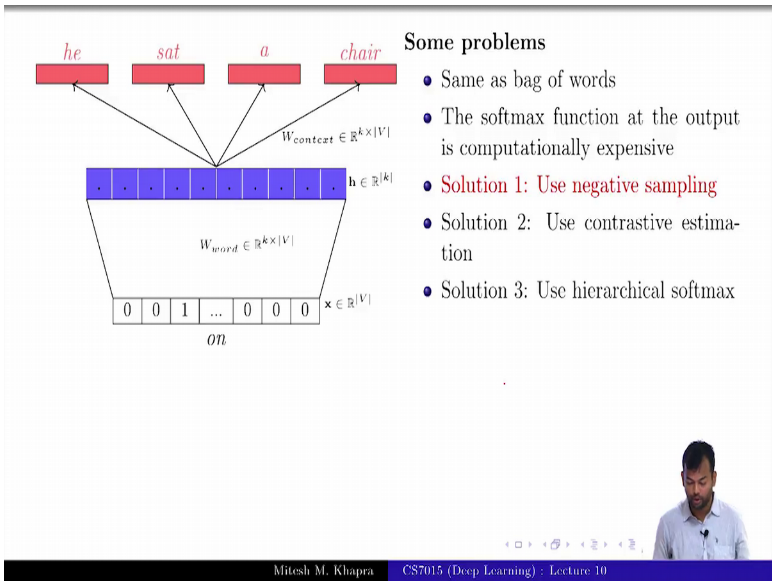
- Notice that the role of *context* and *word* has changed now
- In the simple case when there is only one *context* word, we will arrive at the same update rule for u_c as we did for v_w earlier
- Notice that even when we have multiple context words the loss function would just be a summation of many cross entropy errors

$$\mathscr{L}(\theta) = -\sum_{i=1}^{d-1} \log \hat{y}_{w_i}$$

In the simple case when you are trying to take one word as the input, and only predict one word around it. It just becomes the same as the first case that we saw in the continuous backup words, there is no difference there because there also you take one word and predict the other word right.

So, the entire matt remains the same how many of you get that, and even when we have multiple context words, our loss function is just going to be the sum of the cross entropies for all those D predictions that I need to make or D minus 1 predictions that I need to make. And then once I see a loss function which is a sum of some things I am not worried, because I know how to deal with each of these components and gradients are additive. So, if you have the gradient of some, it is just the sum of the gradients. So, as I long as I know how to deal with one of these I can deal with the sum. So, that is why I do not really worry all of you are at that level.

(Refer Slide Time: 02:40)



he
sat
a
chair
$W_{context} \in \mathbb{R}^{k \times |V|}$
$h \in \mathbb{R}^{|k|}$
$W_{word} \in \mathbb{R}^{k \times |V|}$
0 0 1 ... 0 0 0
$x \in \mathbb{R}^{|V|}$
on
Mitesh M. Khapra
CS7015 (Deep Learning) : Lecture 10

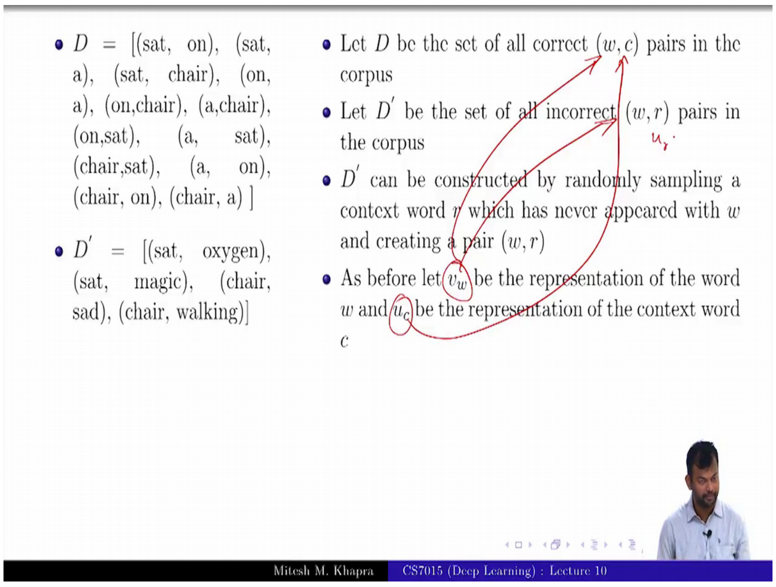
Some problems

- Same as bag of words
- The softmax function at the output is computationally expensive
- Solution 1: Use negative sampling
- Solution 2: Use contrastive estimation
- Solution 3: Use hierarchical softmax

Where you do not worry with the sum as a loss function, what are the problems with this already written there same as the bag of words right.

Now, we are doing these 4 expensive computations at the end. So, the softmax computation is expensive, there are 3 different solutions. And there are 3 different ways that we can deal with it one is something known as use negative sampling, the other is to use contrast of estimation and the third is to use a hierarchical softmax. So, we are going to see all of these and I will shamelessly continue for a few more minutes. So, first we will see use negative sampling because that is very easy.

(Refer Slide Time: 03:13)



- D = [(sat, on), (sat, a), (sat, chair), (on, a), (on,chair), (a,chair), (on,sat), (a, sat), (chair,sat), (a, on), (chair, on), (chair, a)]
- D' = [(sat, oxygen), (sat, magic), (chair, sad), (chair, walking)]

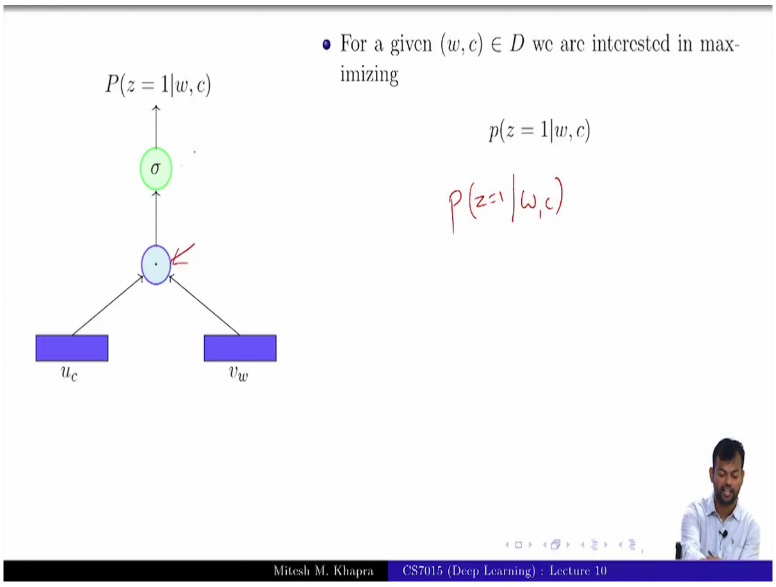
- Let D be the set of all correct (w, c) pairs in the corpus
- Let D' be the set of all incorrect (w, r) pairs in the corpus
- D' can be constructed by randomly sampling a context word r which has never appeared with w and creating a pair (w, r)
- As before let v_w be the representation of the word w and u_c be the representation of the context word c

So, let D be the sat set of all correct w comma c pairs in the corpus, what do we mean by that? All words which actually appeared in the word comma context pair. So, you can look at the vector, which I have constructed. So, sat on sat or sat chair you can imagine that all of these appeared in the context of each other. So, this is my correct corpus as from what I got from my data.

Now, let D prime with a the set of all incorrect w comma r pairs in the corpus, and r here stands for random, some how am I going to construct this corpus. So, I take a word I know all the words which appeared with it and I know all these other words, which have not appeared with it. So, I will randomly sample a word from there and put it as r, is that fine. So, I can compute D prime again D prime comes for free. D was always for free now D prime is also; obviously, for free.

So, I have w comma c and w comma r and I have these corpora D and D prime. And as before let v w be the representation of the word, and u c be the representation of the context word. So, v w will try to these 2 and you see will try to this and u r something else that we will use for this hopefully, is that fine? Okay.

(Refer Slide Time: 04:31)



- For a given $(w,c) \in D$ we are interested in maximizing

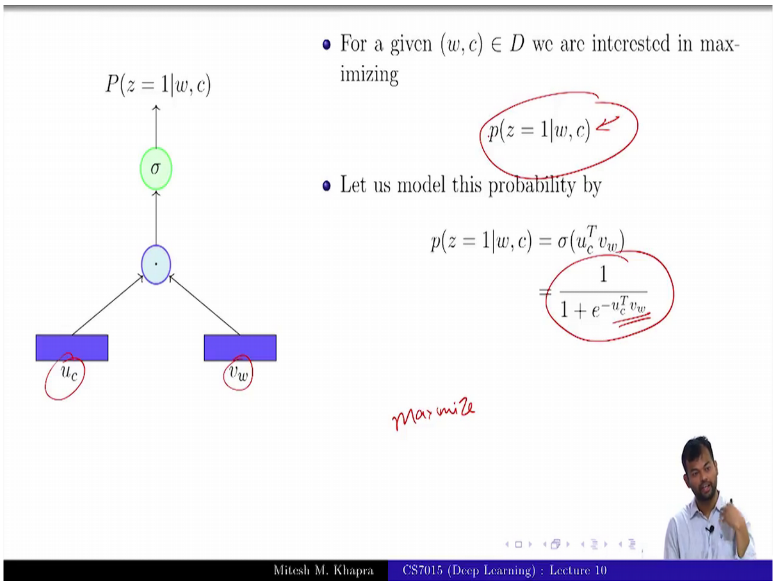
$$p(z=1|w,c)$$

$p(z=1|w,c)$

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

Now, for a given w comma c which belongs to D, which is the true corpus, what are we interested in maximizing. So, let us think of z is a random variable weather which tells us whether this is a true pair or not. So, given w comma c, I want to maximize that p of z is equal to one. Now this is what I want to maximize. Now it depends on me how do I model this probability. So, can you guess how am I going to model this? The answer is there in the figure; can you tell me what is the model that I have chosen? Can you tell me what is the formula for z equal to 1 given w comma c that I have chosen? This stands for dot product this stands for the sigmoid function.

(Refer Slide Time: 05:16)



- For a given $(w,c) \in D$ we are interested in maximizing

$$p(z=1|w,c)$$

- Let us model this probability by

$$p(z=1|w,c) = \sigma(u_c^T v_w) = \frac{1}{1+e^{-u_c^T v_w}}$$

maximize

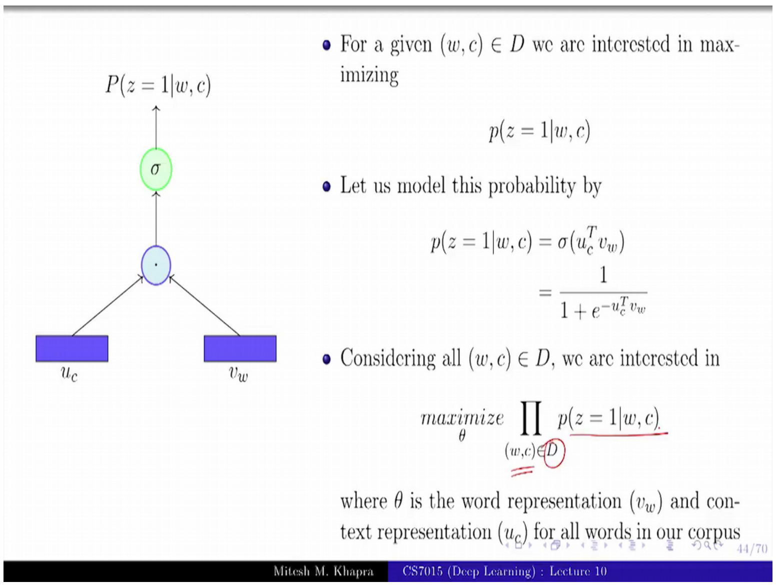
Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 10

I know this is some uc representation, this is some vw representation note that these representations are not learned yet I need to learn them using the training objective that I set.

But at the beginning they are initialized to some random values. And the way I am going to model probability of z equal to one given wc is that I am going to say that, it is just the sigmoid function of the dot product between them how many of you get this are you comfortable with this. This is the modeling choice which I have made or rather the authors of skip gram, right? Now how am I going to now what do I want to do for all w comma c belonging to D. I want to maximize this probability is that fine, for all the w comma c pairs which belong to my true corpus which is the D corpus; I want this probability to be high, how many such pairs do I have many right? So, let us call them as I have n such w comma c pairs.

So, can you tell me what my loss function is going to look like maximize this for the first pair and for the second pair and for the third pair all the way up to the end pairs.

(Refer Slide Time: 06:25)



- For a given $(w,c) \in D$ we are interested in maximizing

$$p(z=1|w,c)$$

- Let us model this probability by

$$p(z=1|w,c) = \sigma(u_c^T v_w) = \frac{1}{1+e^{-u_c^T v_w}}$$

- Considering all $(w,c) \in D$, we are interested in

$$\underset{\theta}{maximize} \prod_{(w,c)\in D} p(z=1|w,c)$$

where θ is the word representation (v_w) and context representation (u_c) for all words in our corpus

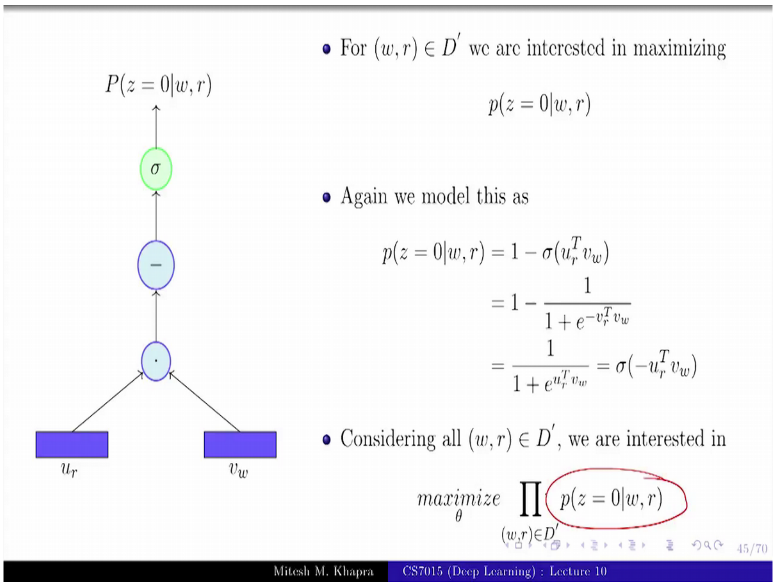
44/70

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

So, what is it going to look like for every w comma c which belongs to my correct corpus, I want to maximize that probability of z equal to 1 given that, w comma c pair right. And since it is an and I will have this product how many of you are comfortable with this.

So, this as such and this is something that you do regularly you should have done this in machine learning or pattern recognition or somewhere right. That you want to basically maximize the log likelihood of the data, which is saying that you want to maximize the probability of every training instance, which is saying that you want to maximize the and of all these probabilities right be you take the and probability is that fine.

(Refer Slide Time: 07:02)



$P(z=0|w,r)$
σ
$-$
$\cdot$
u_r
v_w

- For $(w,r) \in D'$ we are interested in maximizing

$$p(z=0|w,r)$$

- Again we model this as

$$\begin{aligned} p(z=0|w,r) &= 1 - \sigma(u_r^T v_w) \\ &= 1 - \frac{1}{1+e^{-v_r^T v_w}} \\ &= \frac{1}{1+e^{u_r^T v_w}} = \sigma(-u_r^T v_w) \end{aligned}$$

- Considering all $(w,r) \in D'$, we are interested in

$$\underset{\theta}{maximize} \prod_{(w,r)\in D'} p(z=0|w,r)$$

45/70

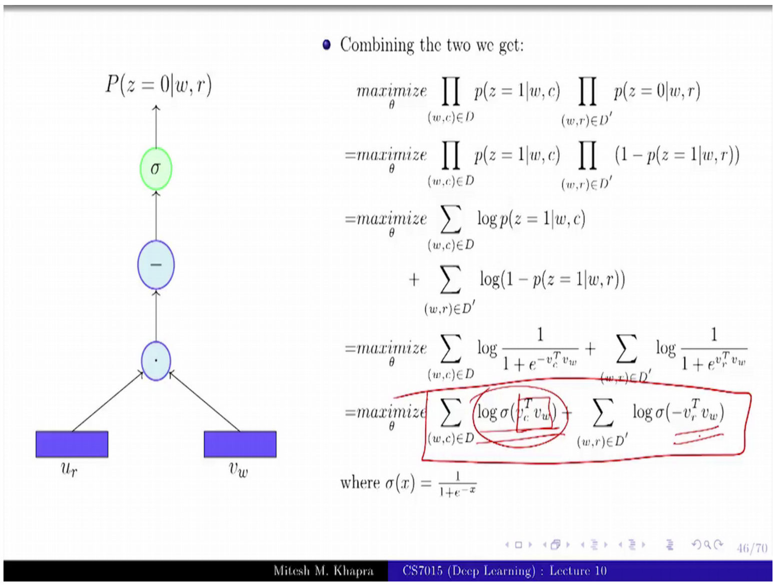
Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 10

Now, for the other case wr belonging to D prime, what is it that I want to maximize? This probability right because I know this is an incorrect pair. So, I want my random variable to output 0.

Now, what is this going to be the probability 1 minus the probability, that it was correct. And that actually if you just simplify a bit it turns out to be this. Now for all the elements which belong to D prime, what is the objective function that I have? I want to maximize this for the first w comma r random pair for the second w comma are random pair and so on for all the random pairs in my corpus. So, it is just going to be a product of all these probabilities, is that fine? So now, what is my total objective function?

For every pair in D maximize that, for every pair in D and for every pair in D prime maximize the 0 probability. So, what is the total going to be is this fine.

(Refer Slide Time: 08:01)



- Combining the two we get:

$$\underset{\theta}{maximize} \prod_{(w,c)\in D} p(z=1|w,c) \prod_{(w,r)\in D'} p(z=0|w,r)$$

$$=\underset{\theta}{maximize} \prod_{(w,c)\in D} p(z=1|w,c) \prod_{(w,r)\in D'} (1-p(z=1|w,r))$$

$$=\underset{\theta}{maximize} \sum_{(w,c)\in D} \log p(z=1|w,c) + \sum_{(w,r)\in D'} \log(1-p(z=1|w,r))$$

$$=\underset{\theta}{maximize} \sum_{(w,c)\in D} \log \frac{1}{1+e^{-v_c^T v_w}} + \sum_{(w,r)\in D'} \log \frac{1}{1+e^{v_r^T v_w}}$$

$$=\underset{\theta}{maximize} \boxed{\sum_{(w,c)\in D} \log \sigma(\boxed{v_c^T v_w}) + \sum_{(w,r)\in D'} \log \sigma(-v_r^T v_w)}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$

46/70

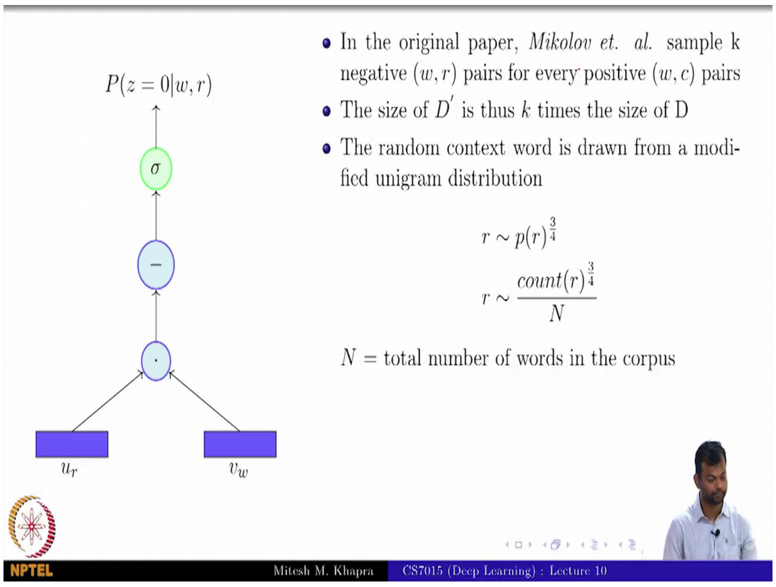
Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 10

How many of you agree with this? So, for everything belonging to D I had this and rule for everything belonging D prime, I had another and rule and I am interested in both the acts right maximize for D and maximize for D prime of course, different quantities for D and D prime fine. So, you get this once you basically take the log and so on. So, this is a simple set of math operations that I do you will end up with this neat formula.

That for all the w comma c pairs belonging to D, you want to maximize this quantity which means; you want to maximize what you want to maximize the when will this quantity be maximized. When the dot product between the 2 is high; that means, again what are you doing? We are trying to bring the context vectors close to the word vectors again transitively what will happen the words, which appear in the same context will go close to each other. What is the additional thing that you are ensuring here? The words which do not appear in the same context you are trying to push them apart, why? Because of second loss function.

You see the difference between the 2 now. In the first case you are only opt I mean you are obsessed with bringing things close together, here you are also focusing on the case that where you do not want certain things to be close together because they never appear in the same context that fine. So, you see that this is a more powerful loss function in the earlier one. So, that is what the skip gram model does.

(Refer Slide Time: 09:27)



- In the original paper, *Mikolov et. al.* sample k negative (w,r) pairs for every positive (w,c) pairs
- The size of D' is thus k times the size of D
- The random context word is drawn from a modified unigram distribution

$$r \sim p(r)^{\frac{3}{4}}$$

$$r \sim \frac{count(r)^{\frac{3}{4}}}{N}$$

N = total number of words in the corpus

And in the original paper Mikolov et al sample k random pairs for every positive pair right.

So; that means, if your size of D was n what was the size of D prime b k into n. So, that they had that many positive examples and k times that the number of negative examples. And this was a hyper parameter which was tuned and they used a value of k such that it gave them the best results. Also remember that we have this problem of constructing w comma r. Now I said that consider all the words which do not appear with your word, and sample from there and put something there. So, they used a slightly that means, how do I sample that? One is I assign all the words a uniform distribution that every word is equally likely, what is a better way of doing that? Okay I think I just finished this next time.