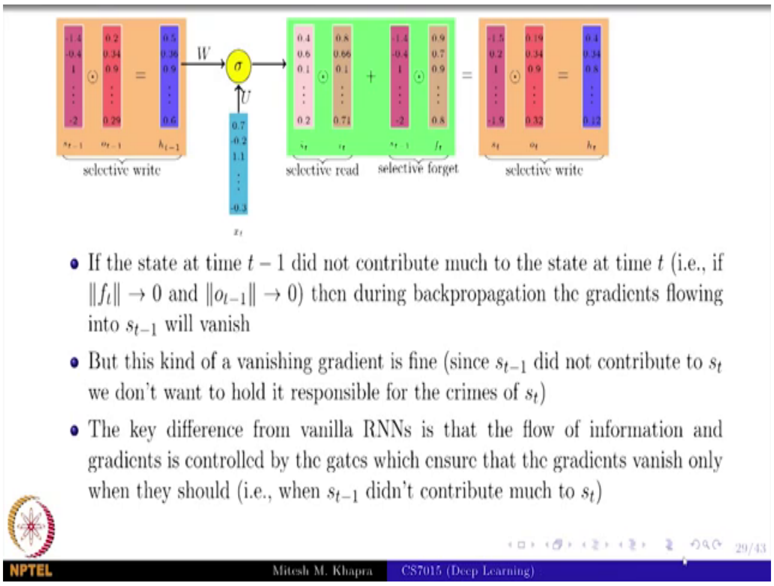
Deep Learning
Prof. Mitesh M. Khapra
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 111
How LSTMs avoid the problem of vanishing gradients (Contd.)

(Refer Slide Time: 00:12)



Ok so, will start from where we left off. So, in the last class we started with this motivation that recurrent neural networks have this problem of Vanishing and Grade Exploding Gradients. And we wanted to arrive with some principle way of avoiding this. So, you have first started with this intuition that in many real life situations like for example, the human brain or the whiteboard. We tend to these to these three operations called selective read, selective write, and selective forget. And they essentially help us in dealing with these finite sized memories right or whether it is a whiteboard which is finite sized or your brain or whatever it is right.

So, can we is it possible to kind of improve RNN's which also suffer from this problem that they have this finite sized memory. And hence if you are trying to capture everything from time step one then by the time you reach time step t where say t is 30 or 40 or so on. It is quite natural that whatever you have learned earlier will get move off to an extent that it just is not recognisable anymore right.

So, you wanted to deal with this problem and with that we motivated selective read write and forget. And then we introduced some equations or converted this into a model and this is the diagram that you see is the model actually that is the L st m cell data. And it has these three gates output gate, input gate, and forget gate and which perform these three functions of selective read write and forget. So, intuitively all these was fine, but we need to be more technical in terms of you trying to deal with a problem of vanishing and grade exploding gradients.

So, how does it solve that problem all that makes or the story seems fine, but how does this actually relate to the math. So, we saw some intuition for that and the intuition hinsed on this observation that. During forward pass the gates control how much of information passes from one state to another. And in particular if you have the situation that from one time step to another say the forget gate tells you that keep forgetting point 5 of the previous state. Then by the time you reach say the 100 state you would have forgotten 0.5 raise to 100 of the first state.

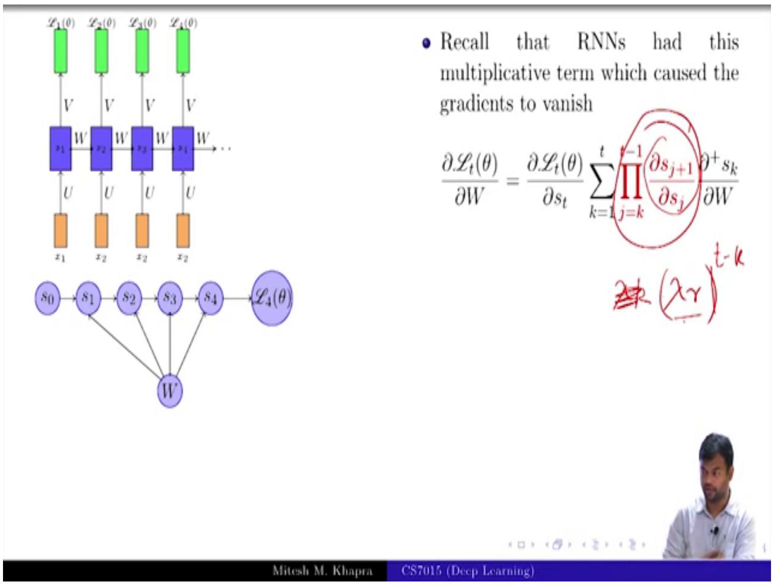
So; that means, even during forward pass the information from state 1 vanishes. So, if it vanishes during backward path that is also fine, because state 1 did not contribute to state 100. And that was the intuition that all this hinsed on now we are not going to do much different from this intuition. We just going to see the corresponding equations for these intuitions and just make a more I would not call it rigorous but more mathematical proof on why L st m solve the problem of vanishing gradients.

And we are also sure that they actually do not solve the problem of exploding gradients and then we will see a simple trick of dealing with exploding gradient. That is what we will do in the remainder of this particular lecture and then will move on to the next lecture in this lecture.

(Refer Slide Time: 03:02)

So, we will now see an illustrative proof of how the gates control the flow of gradients right.

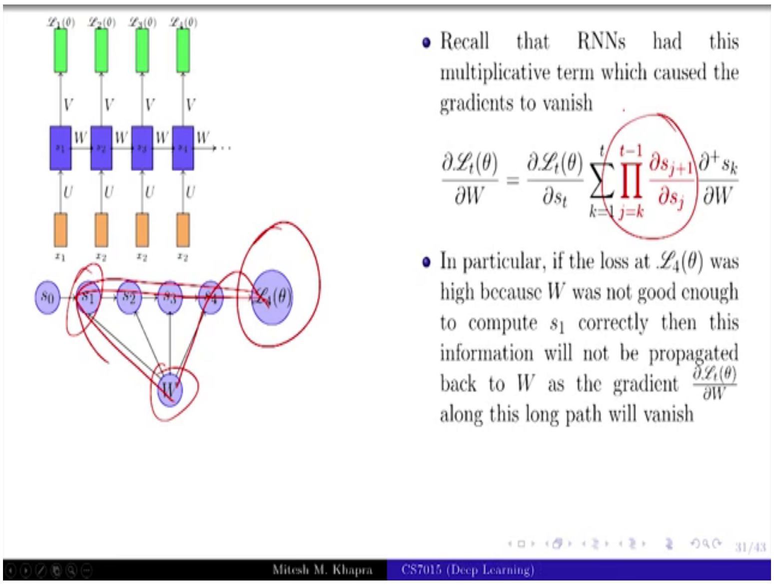
(Refer Slide Time: 03:11)



So, we call that this is the control this is the flow diagram or the dependency diagram that you had for RNN's. And in particular because you are dealing with an ordered network we add this explicit and implicit derivatives and finally, you came up with this multiplicative form. And this term here is actually a matrix because it is a derivative of a vector with respect to a vector.

And then this same matrix was getting multiple times and then we did this proof it showed that this term is actually lambda gamma ok. It is actually proportional to this term right and as if lambda into gamma is greater than 1, then this will explode. If it is less than 1 then it will vanish given sufficient times that is.

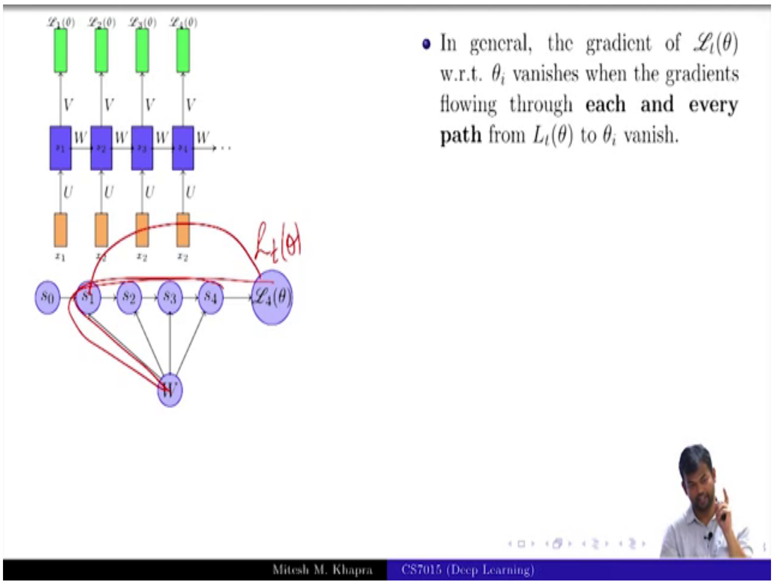
(Refer Slide Time: 03:56)



Now, in particular what is happening here is the following that you have this loss at time step t you have the time step is 4. Now, what if this loss or this error occurred, because W was not good enough to compute a good value for S 1 right. So, W was at a certain configuration based on that you computed S1. And that S1 was not good enough which eventually led to the error at time step 4 all of you if you can imagine this situation that you mean you not being not be able to do something well at S1. Now this needs to be told to W so that it can improve right. And that information has to come through S 1 that information is already going from here, but this information is about how badly it performed in computing S 4.

This is not how badly it can perform in computing S 1. So, that information has to travel to W all the way through S 1 and that was not happening because this path do not look at the bullets this path was actually vanishing. And that is what this multiplicative term says that as the number of times that increased that time that path would vanish ok. So, that is the actual problem that we are trying to deal.

(Refer Slide Time: 05:05)



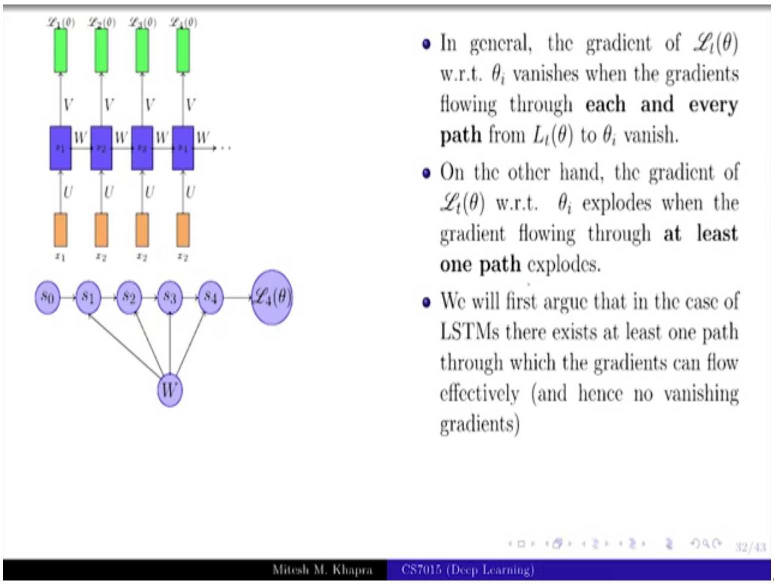
So, now what is the general situation here right, the general principle is that the gradient of L theta at particular time step say here we are considering L 4 so I will just call it L t with respect to any parameter theta i. The parameters at W u v b and c with respect to any parameter it would vanish if all the paths leading to that parameter if it vanishes. So, with respect to this particular path so that is the only path which leads to W through S 1. If there were multiple paths if there was say one such direct path right if we had you some other kind of connection which gave us this direct path then it would still have been fine.

But there was only one path leading to W through S 1 at the gradient vanishes along that path then the gradient will vanish ok. If there were multiple paths then only if the gradient varnishes across all the paths then the gradient would vanish is it fine. What is the corresponding rule for exploding gradients? If there are multiple paths the gradient would explode if.

Student: (Refer Time: 06:06).

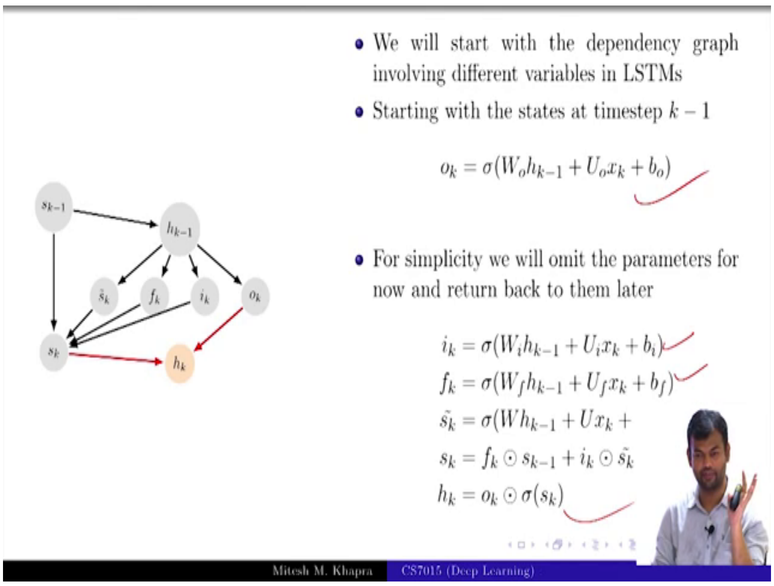
If it vanishes through any o[ne]- if it explodes though any one of the paths ok.

(Refer Slide Time: 06:11)



So, these are the two things that we need to consider ok. So, to prove that in the case of L st m this does not happen. For the first case will have to show that there are at least one path through which it does not vanish and for the second case because we are going to show that it explodes we just have to show that there is at least one path through which it can explode ok. So, these are the two things that we need to prove and the first thing that we are going to focus on; is the vanishing gradient problem.

(Refer Slide Time: 06:38)



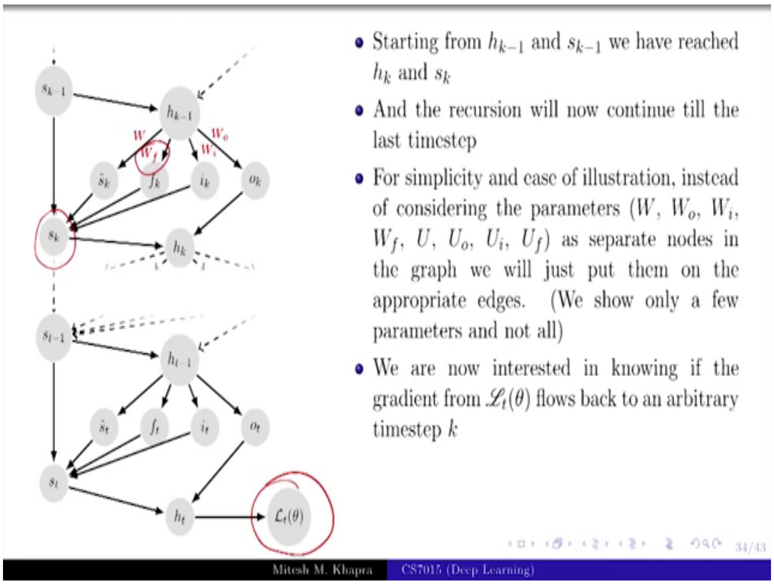
So, will start with the dependency graph for LSTM's; that means, I want to draw something similar for LSTM's involving all the different elements in LSTM. So, what are these different elements the two rhyming things one being gates states ok. So, gates and states that the two things that we care about. So, let us look at all these. So, starting with states at time step k minus so, at time step k minus 1 you have this two states sk minus 1 and h k minus 1 ok. Using h k minus 1 you are going to compute the output gate at time step k. And it is also depends on these parameters W o, U o and b o right which is obvious from the equation.

Just to make sure that this diagram remains tractable, I am going to get rid of the parameters and I will come back to them later. So, right now will just focus on the states and the gates ok and then you have these other intermediate states and the other gates right. So, you had f k you had i k. So, add these three gates the temporary state and then what else what are the other two things at time step k. So, we saw this diagram about all the computations which happen at time step k right. How many computations happen? Three states and three gates right.

So, you seen the three gates and this one temporary state. So, which are the other two things? There is no selective forget with you guys is early everything forget. Hint look at the grey cells and change the time step. What will you get away are you all I mean we did LSTM's two days right I mean are you all with that or should I we need to revise something; mean I do not need to revise it, but we going to is it fine ok. So, sk and the other thing hk remember that sk also depends on h k just stare at this for 30 seconds and make sure that you are with it right. All the equations are there these are the see 6 equations that or the 6 computations which happen at time step k.

There are three gates and three states and the dependency graph is obvious from these equations; except for the fact that I have ignored the parameters. How many if you are comfortable with the equations and the graph corresponding graph please raise your hands high. So, I think it should be right we have these six equations and we have this dependency graph.

(Refer Slide Time: 08:50)



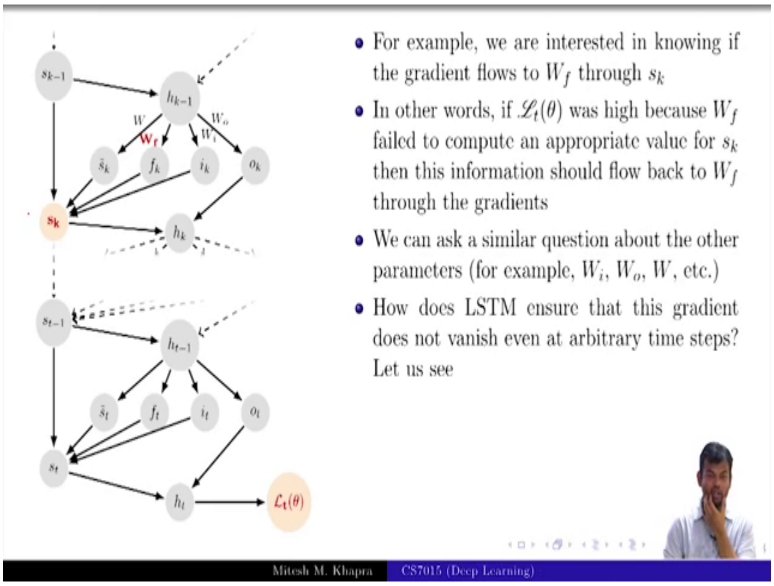
Now, starting so, what happened in the graph is, we started from sk minus 1 and h k minus 1 and we reached sk and h k which were the outputs at the next state. Now what will happen from here? We were looking at recurrent neural networks recursion is the answer. What will happen now?

The same graph will keep recursing right for the next time step and up to the last time step right, does that makes sense ok? This is much more complicated than the dependency graph that we had for RNN's right by just because there are so, many we in RNN we just had this one state and no gates so here but we have these three states and three gates that is why this so many paths ok. Now for simplicity what I will do is, I will not draw separate nodes for the parameters all the in the case of the RNN dependency graph I had drawn them separately. What I am going to do is, I am just going to put the parameters on the corresponding edges right.

So, f k actually depends on W f, it also depends on U f, and it also depends on that bias. But I am just going to take a small set of parameter I am only going to focus on the W's not the U's and the biases ok. There is only for illustration for no other reason right and whatever arguments or proof that we are going to see it holds for all the parameters, but we just need to prove it with respect to one parameter and the same story repeats for everything ok. So, this is the dependency graph and these are the parameters. Now what I am interested in knowing is that, there was some loss at time step t and maybe that loss happened because W f was not good enough to compute s k.

Of course W f computes f k and then f k helps in computing sk, but maybe W f was not I am just short it short circuiting it and saying that W f was not good enough to compute sk right. And that is why I want the gradient to reach to W f through this sk that is what I want do ok.

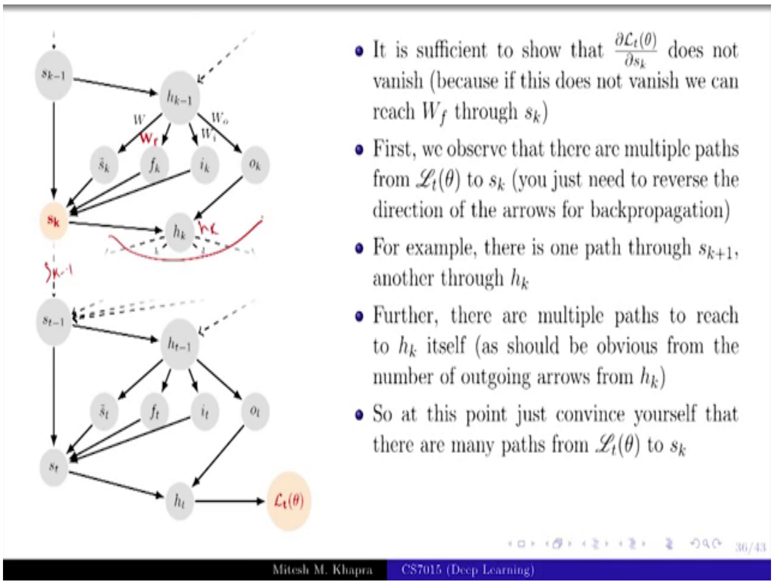
(Refer Slide Time: 10:42)



And this exactly what I said I am interested in knowing that if this loss can reach Wf through sk right. So, all the three highlighted things that what I am interested in, I am interested in the path to W f through sk. Of course, there are many other paths to W f, but they do not account for the problem in sk; is that fine everyone is clear the setup ok?

Now and we can ask similar questions about all the other parameters the W's the us the the input gate parameters the output gate parameters and so on right. There is nothing so special about W f the same question holds for all these other parameters also ok. Now how does LSTM ensure that this does not vanish? So let us see that.

(Refer Slide Time: 11:20)



As I argued earlier it is sufficient to show that this gradient does not vanish ok. If I can show that this gradient does not vanish, then I am pity sure there is only there is no recursive connection here because it just a single connection. So, there is no recursive connection here. So if I can show that the gradient reaches up to this point, then after that I can be sure that it is going to reach W f everyone buys that set up right that is what I need to show?

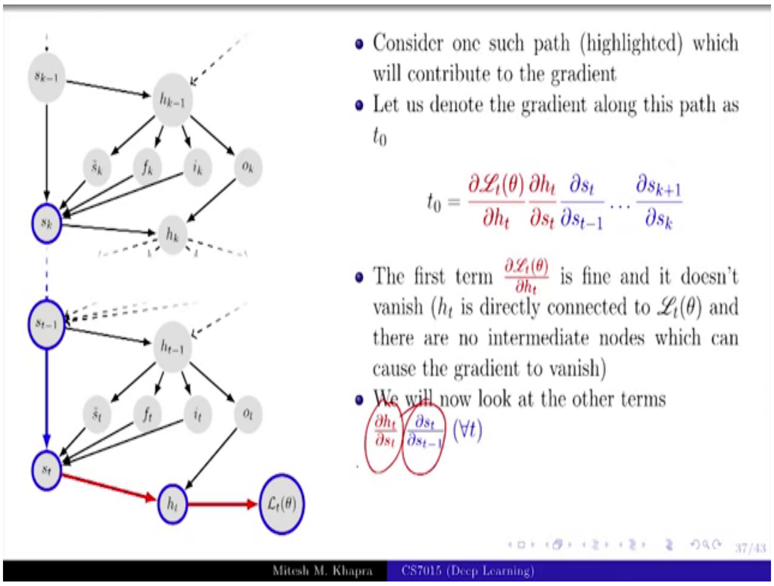
So, to prove that the gradient reaches W f I just need to show that it reaches sk that is the only thing that I need to show. And the first thing I am going to observe is that there are multiple paths to reach to sk which are these paths? One through sk plus 1, because sk contributes to sk plus 1 the other through.

Student: H k.

H k which is visible, but now also notice that how many paths are there to reach h k itself 4. Not four actually that is going to be combinate relate because there four outgoing edges from here, but then again there will be four next stage and four next stage and so on right. So, let us not count the number of paths, but let us just convince ourselves that there are many many paths to reach to sk from L t theta. Everyone is convinced about that we are not counting the exact number of paths that is not very hard to do. But all we are saying is that we know that there is one path through sk plus 1, one path through h k and h k itself seems to have many incoming path during back propagation.

So, there are many many paths which are reaching from L t theta to sk. Everyone is convinced about that anyone who has a problem with that? Now to show that the gradient does not vanish what do I need to show of all the paths the set there exist at least one path through which the gradient can flow that is what I need to show ok. Even if I vanishes across all the other paths I am still fine with it ok.

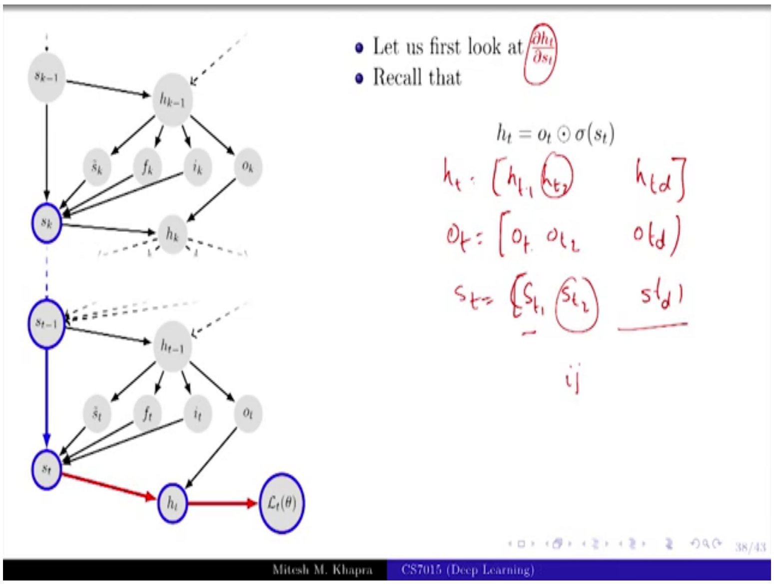
(Refer Slide Time: 13:05)



So, now consider one such path which is this highlighted path that is a valid path to reach to sk. Now let us denote the gradient along this path to be t naught and the total gradient is going to be a sum of many such paths right. So, I am calling this path as t naught and this is what the gradient look like ok. So, this is simple just this red path the next red path and then the series of problematic multiplications right you have this recursive multiplications again. So, everyone agrees that red is good, the red path there is no recursion the gradient will flow right we just need to focus on the blue path everyone is convinced about that right ok.

So, that is good the first term is fine as I said because it directly connected to L t there is no recursive or no other intermediate nodes. So, the gradient will just flow through that there is not a problem there and now we look at the other terms which is first is d h t by st and the other is this ok.

(Refer Slide Time: 14:03)



So, let us look at dou of h t by st what is this going to be? Tensor, vector, matrix scalar at this point in the course I want a unanimous answer.

Student: Matrix

Matrix right and recall that in particular the equation was of this form ok. So, what is the derivative going to look like even without computing can you tell me something profound about it; it will be a dash matrix. Big matrix, how many if you say diagonal matrix? How many if you do not think it is a diagonal matrix please raise your hands total sum is never one. So, remember that h t is equal to h t 1, h t 2 up to h t d. And you have o t equal to o t 1 o t 2 o t d and st equal to st 1 st 2 s t d. So, h t 2 depends only on o t 2 and st 2 right it does not depend on in particular does not depend on any of the other st's.

So, we have already seen this before in such cases what is the i jth entry of this matrix of the gradient matrix derivative of h t i with respect to st j which of these terms are going to be zero wherever?

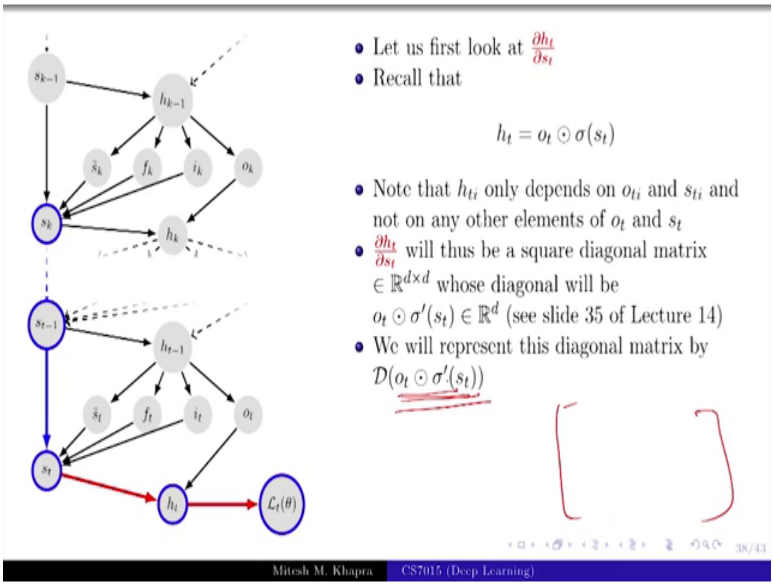
Student: I not equal to 0.

I is not equal to 0; that means, it results in a.

Student: Diagonal matrix

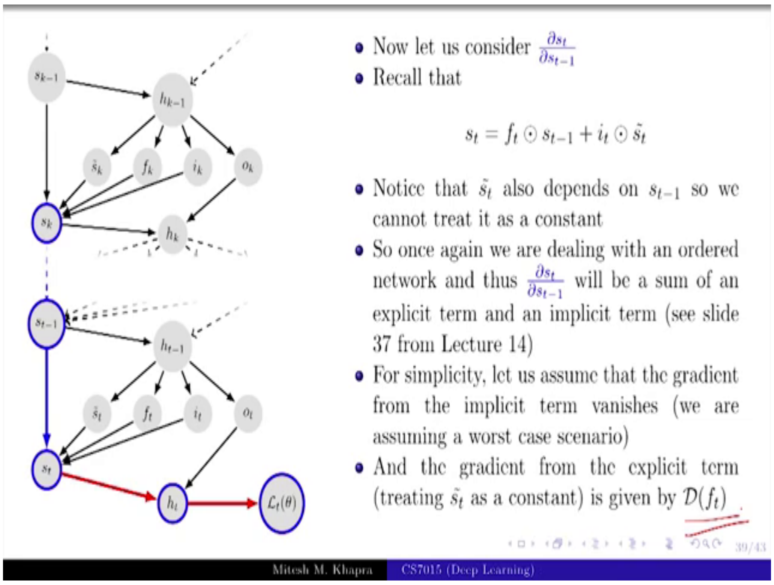
Diagonal matrix.

(Refer Slide Time: 15:31)



So, that is exactly what is written here and the diagonal elements are going to be this is that fine everyone with this ok. So, now, this diagonal matrix which contains this on the diagonal I am going to represented by the following notation is that fine. So, this is a diagonal matrix where every element is, I mean this is actually a vector right everyone agrees this is a vector. So, this diagonal is this vector is along the diagonal of this matrix how many if you get this notation? If you do not get this you will not understand anything else.

(Refer Slide Time: 16:07)



Now let us consider dou st by dou st minus 1 this is what st is equal to. So, what is the derivative of dou st by dou st minus 1? F t right f t right what else, why no, why are you rebelling, what the I mean st only right. If it is can you treat this as a constant no why? Because this is a dash network.

Student: (Refer Time: 16:34).

So in an ordered network the derivative will have.

Two terms which are those?

Student: Explicit.

Explicit and implicit In the explicit term what you assume? The other terms to be a constant right fine. So, st I mean S tilde t also depends on st minus 1. So, we cannot treated as a constant so once again this derivative is going to contain an explicit term and an implicit term. Now I am going to make a worst case assumption. I making this assumption I making this assumption that actually the implicit term vanishes. Notice that this not favourable to me I am trying to prove that the gradient does not vanish the gradient is a sum of two terms I am saying it let the worst case be that one of these terms vanishes ok.

So, this is not a favourable assumption this is a unfavourable assumption which I am making. So, let us fine. So, I making the assumption that the implicit term vanishes. So, what is the explicit term actually?

Student: F t.

F t and what kind of a matrix is that?

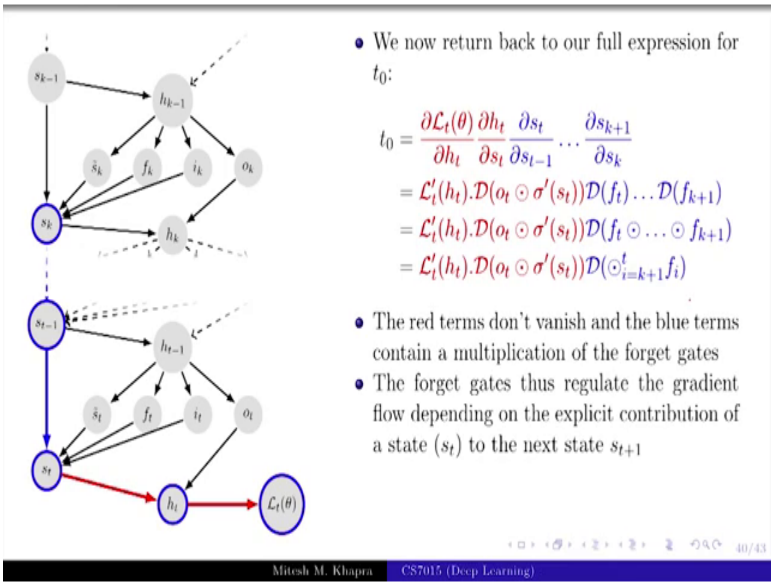
Student: Diagonal matrix.

If you agree that it is a matrix first of all. It is a diagonal matrix again and what is the diagonal?

Student: F t.

F t right. So, I am going to represent it as D of f t with that fine?

(Refer Slide Time: 17:46)



So, remember that the original equation ah had three terms all of these the last blue once for all identical. So, this is not problematic because this is a directly the last layer this we have already derived a form this is sum diagonal. And now for each of these we have a form; do you get that these are the three paths that we have done so far. So, let me just substitute them, this is what it looks like ok. Now this is a product of diagonal matrices what will the product look like?

Student: Diagonal matrix.

A diagonal matrix and each element would be each element on the diagonal would be a.

Student: Product

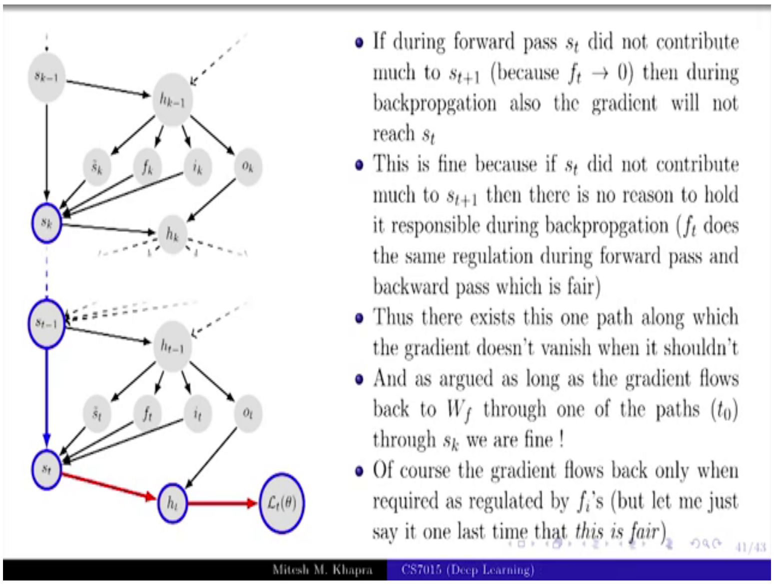
A product of all those things right. So, is it fair if I write it as this right which I can write it as this ok. Now just stare at this equation and tie it back to the intuition that we developed something about the gates regulating the flow of information you have a multiplicative term here right. Whenever there is a multiplicative term we have a problem, because remember these gates are between 0 to 1.

So, there is a chance of vanishing agencies that? You are multiplying t terms all of which are between 0 to 1. So, there is a chance of vanishing. But I am going to end this proof by saying that the gradient does not vanish. So, by what am I going to do now ok. I make the statement the gradient could vanish, but this kind of vanishing is fair what do you mean by that now when will the gradient vanish?

Student: Product

At this product of the forget gates vanishes, but if the product of the forget gate vanishes; that means, what would have happened during the forward pass that information was not carried all the way back, all the way front two times step t right do you see that ok. So, that is the main reason here right. So, the red term does not vanish, the red term time zone vanish the blue term can vanish, but it will vanish only if during the forward pass also this multiplicative term at cause the information to vanish by the time you are reach the time step t. how many if you get this and this exactly what I meant earlier by saying that.

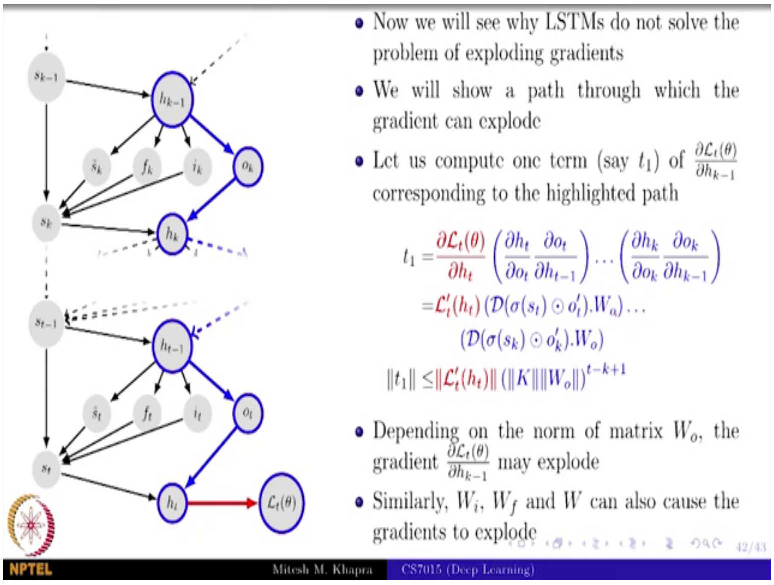
(Refer Slide Time: 19:45)



If during the forward pass st did not contribute much to st plus 1. Because the forget gate was tending to 0. Then during backward pass there is no need to pass this information back to st right because during forward pass you did not contribute. So, during backward pass why should I hold you responsible right? And this is absolutely fine to do this and this is exactly what the equation tells us that they gauze the gradient will vanish only if things vanished in the forward pass ok. And the gates are doing the same regulation in the forward pass as they will do in the backward pass so everything is fair is thatok?

And does there exist one path along which the gradients will not vanish when they do not need to vanish. So, if during forward pass all the gates were on; that means, the information from state one was actually carried all the way up to state t then during backward pass what will happen? The information will go all the way back right is that fine. So, the gradients flow back only when required as regulated by the forget gates and this is fair because if you are regulating the same thing in the forward as well as the backward direction then you are not doing anything wrong ok.

(Refer Slide Time: 20:51)



Now that is a proof for LSTM solve the vanishing gradient problems or in other words the gradients vanish only when required and not unnecessarily or arbitrary as is to happen in the case RNN's. Now we will show there exist one path along which the gradients can explode right. So, let us show that path. So, consider this path now this path is again also active so I if we consider the path to h k there is going to be active for all the gates and all the states right. So, in whatever gates or states you are considering this paths would be there.

And this is what this path looks like you have the derivative with respect to the last layer and then you have these guys ok. These pairs h t by o t o t by h t minus 1 and so on again fine with this so, far what is the derivative of h t with respect to o t? We do not remember the equations. So, I will just tell you directly. So, based on whatever we have done so, far just trust me that this is what each of the terms in the bracket looks like. We can go back and check this is just comes directly from the equations. Now what is happening here does this look very similar to the situation that we had with RNN's.

We had a diagonal matrix and a weight matrix and a repeated multiplication of these right and again the diagonal matrix is bounded the weight matrix is bounded. So, now, the repeated multiplication could explode is that fine. So, it does not solve the problem of exploding gradients. But it solves the problem of vanishing gradients, but now still this is bad for us right whether the gradients explode or vanish our training is going to get messed up. So, how do we deal with this for exploding gradients what will you do?

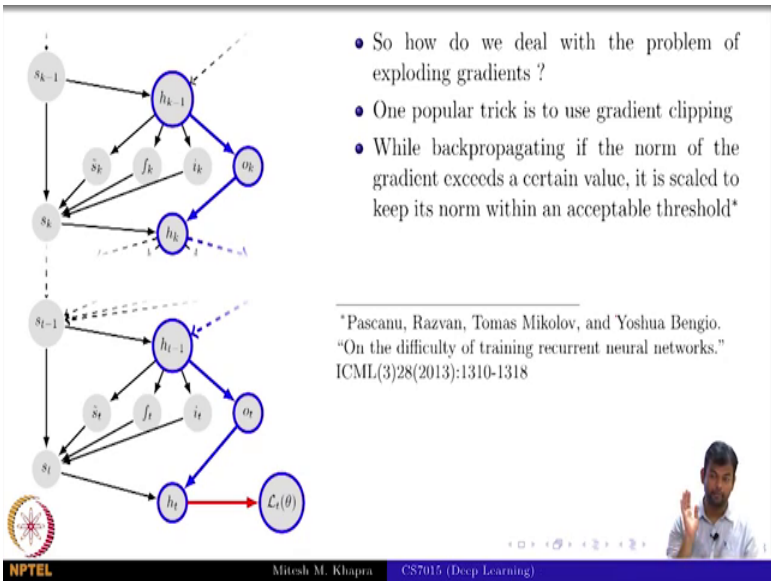
Student: (Refer Time: 22:27) clipping.

We will do?

Student: Clipping.

Clipping right.

(Refer Slide Time: 22:32)



So in fact, is the way of dealing with this is gradient clipping if the norm of the gradient exceeds the certain value, when we are going to just clip it to a certain threshold. And this is fine because we care about the gradients only for the direction and not for the magnitude. Anyways when we introduce a learning rate, we are doing some kind of scale down for the gradient magnitude. So, this is just being more explicit and being careful that if the gradients are non manageable in terms of their magnitude.

Then we just going to keep them to some manageable value while being faithful to the direction. And the direction is what? Matters so that is why exploding gradients is easy, but in the case of vanishing gradients you do not have direction also because the entire gradient becomes 0, so there is no direction there right. So, that is why vanishing gradients is more serious than exploding gradients. And as long as LSTM solve that they are fine with it is that fine ok. So, that is the end of this lecture.