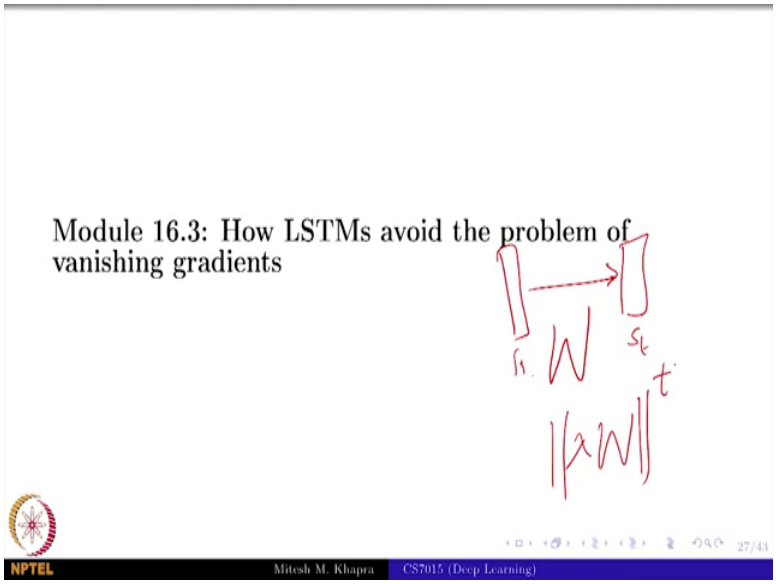
Deep Learning
Prof. Mithesh M. Khapra
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture - 110
How LSTMs Avoid the Problem of vanishing gradients

So, that was LSTM and GRUs.

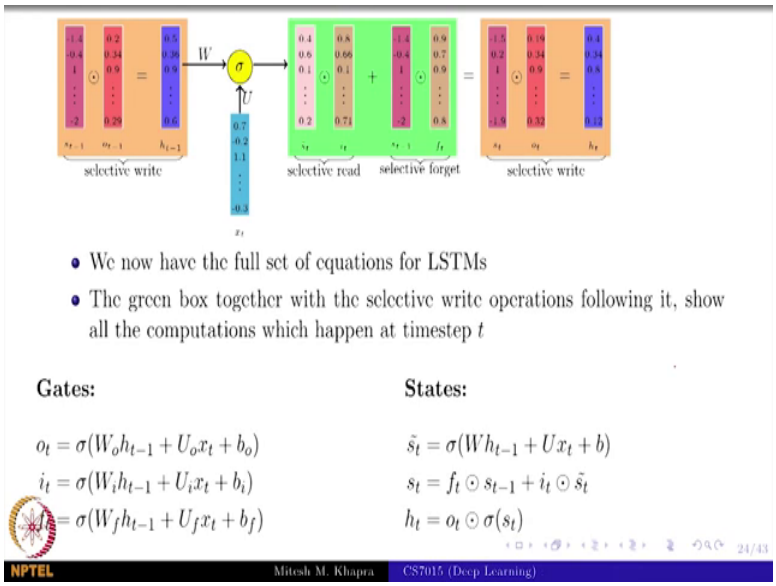
(Refer Slide Time: 00:15)



Now, the issue is that, I have given you a very explanation that why you selectively read write and forget should work, but you have not actually formally proven or even given an intuition for with these sets of equations, how are we sure that the gradients will flow back right. We introduced a bunch of equations, remember in the case of LSTMs sorry in the case of RNNs, the problem was because of the recurrent connections right. Because you had these recurrent connections this W which was the recurrent parameter right, which was connecting cell state S t minus 1 to cell state S t.

This was repeatedly appearing in your gradients right and that was causing the problem because, when you had this multiplicative factor lambda into W and then if you compute the and this was lambda raise to t. So then if you compute this magnitude then if the magnitude of W blows up then the whole thing will explode, if the magnitude of W vanishes then the whole thing will vanish right. That is the problem that we had.

(Refer Slide Time: 01:29)

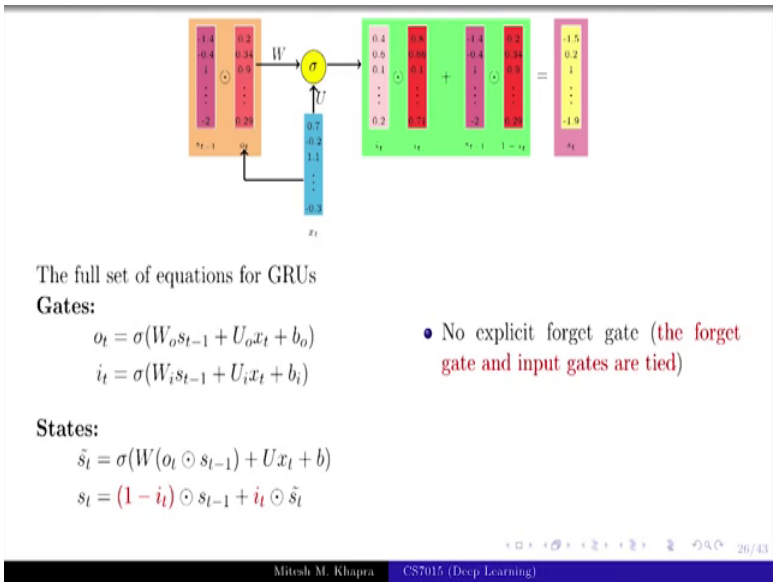


So, this was because of the recurrent connections. Do we have recurrent connections in LSTMs or GRUs for that matter? Do you have recurrent connections yes or no?

Student: Yes.

Yes, so then that problem could still occur right, I mean if you had that the crux of the problem for the vanishing gradient was this recurrent connection which is getting multiplied. And hence reading to problem, but we still have recurrent connections the case of LSTMs also and why should things become any easier in this case. How many if you get the question, how many if you can give me the answer selectively. That is a good answer. So, can you think of what is happening here, so first thing that we going to do now so I will go on to the next module.

(Refer Slide Time: 02:05)



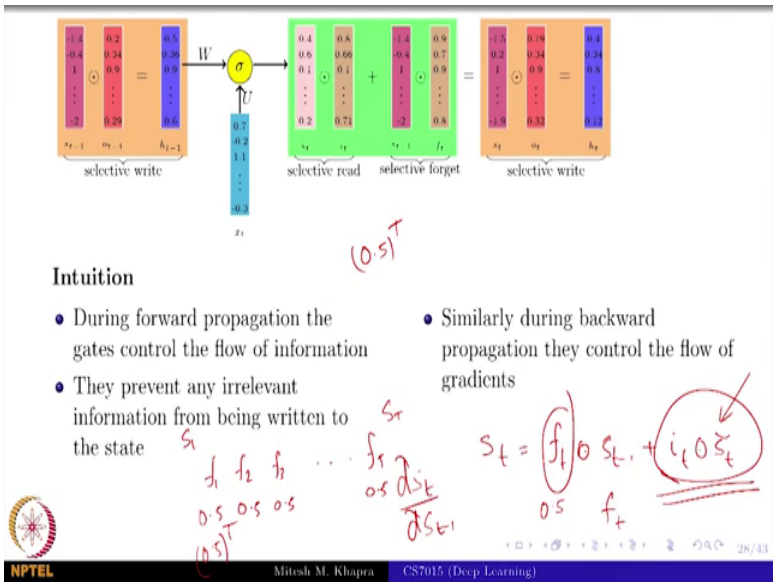
When I going to give you intuition for what is happening and then we will do slightly, in fact, a rigorous proof of why it actually solve the vanishing and exploding gradient problem ok. So, let us look at the intuition first. How LSTMs avoid the problem of vanishing gradients, I am only focusing on vanishing gradients exploding gradients are actually easier to deal with. Why?

Student: (Refer Time: 02:26).

What can you do, what are we interested in when we compute a gradient direction. So, if the magnitude is very large what can we do, just normalize it and restricted to be a certain magnitude, so that is known as gradient clipping. So, exploding gradients in that sense is still not a big problem, but vanishing gradients is because, if it vanishes you cannot do anything, because you could think of it that you already have a learning rate which is getting multiplied with the vector the gradient. Now in addition to the learning rate which was anyways clipping the norm of the gradient right. So, you are doing an expressive clipping also.

So, it just like a additional learning rate inductions right ok.

(Refer Slide Time: 03:06)



So, here the intuition and then will go to the more rigorous stuff, not in this class probably. So, during forward propagation, the gates control the flow of information right. The gate decides how much of S t minus 1 should be pass to S t ok. And they prevent any relevant information from being returned to the next state. Similarly during back propagation, the gates will regulate the flow of information. So, what I mean by that is that if at a certain state, you have computed S t is equal to f t into S t minus 1 plus i t into s till the t right.

So, this gate is actually deciding how much information flows in the positive direction ok. And suppose this gate value was 0.5, so the 0.5 of this information from S t has been carried on S t minus 1 ok. Now during back propagation what is the derivative of S t with respect to s t minus 1 going to be partial derivative is going to be a f t. Think of S t and S t minus 1 as single variables like you know n dimension variables, then the just f t. Of course, you are forgetting that, what kind of a network is this ordered network right. So, you cannot read as till de t as a constant.

Where s tilde T also somewhere depends on S t minus ok, but just this assume that, maybe this vanishes and that is the worst case assumption right. Because, I do not want it to vanish, but I am assuming that the second term vanishes, but even then with the first term I will have a gradient which is proportional to the gate. Why is that fine? So, remember that I am not making a easy assumption, I am making a worst case

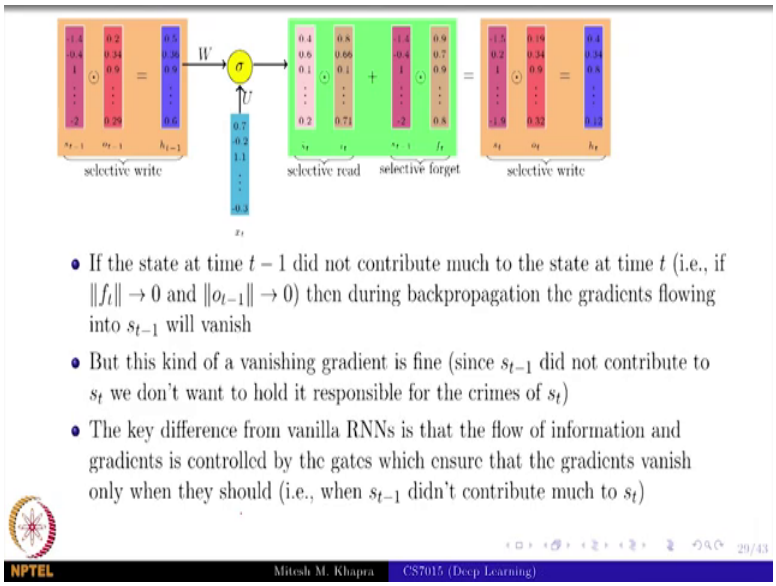
assumption. This is not favourable to me, I am saying that the second term vanishes and i don ot want it to vanish, but i am just trying to prove that even in the worst case, by the second term vanishes, you still have this gradient f t from the first term right.

And why is that good, why is that ok, because f t decides how much flow in the forward direction. And it is also deciding how much goes back in the backward direction. So, it is a fair regulator with says that if I passed on only this much information in the forward direction. Then during backward pass also I should only make a responsible by this much now let us look at a situation where you had f 1, f 2, f 3 upto f t and all of these gates were 0.5. Now 0.5 implies a reasonable value, but when we have 0.5 raise to t and t is a large value, what is going to happen, this quantity is going to vanish.

So, what is happening is that S1 contribution to ST in the forward direction itself had S1's contribution to ST in the forward direction itself was had already vanished right because, it was continuously getting multiplied by 0.5, 0.5, 0.5, so it is like this Chinese Vespring problems right. So, this guy said something whereas, next guy added noise, the next guy again added noise and so on, till the time it reach the T-th guy this information was completely lost So, in the forward pass if S1 did not contribute to S T in the backward pass should I make it responsible for the crimes of ST, no.

So, what is happening in the backward pass, again the gradients are getting regulated by the same forget gates. So, again in the backward pass will have a situation that, by the time the gradient reach is S1 it would be 0.5 raise to T and that is fine it is going to vanish, but that is because even in the forward pass it vanished. So let it vanish in the backward pass also. So, this kind of vanishing is ok.

(Refer Slide Time: 06:41)



So, this is just the same thing written in words. So, if the stated time t minus 1 did not contribute much to the state at time t because, f t was tending to 0 right. Then during backpropagation the gradients flowing into s t minus one will also vanish because again during backpropagation the gradients will get multiplied by f t and they will vanish.

But this kind of vanishing gradient is fine. This is fair because, if we did not contribute in the forward direction why should I help you hold your responsible in the backward direction right. So, that is fair. So, the key difference from RNNs is that the flow of gradients is now controlled by gates, which give the same regulation in the forward pass as well as the backward pass right. So, only if you contributed to something you will be held responsible. If your contribution vanished your responsibility in the backward pass will also vanish right. So, that is the intuition.

(Refer Slide Time: 07:27)

And will next see an proof for this. A proof actually it s as based on the intuition, but I just make it more formal in terms of introducing the notations and so on. So that problem we will do it in the next class ok.