Modern Computer Vision

Prof. A.N. Rajagopalan

Department of Electrical Engineering

IIT Madras

Lecture-32

So, what is done is you know there is a different architecture, so RNN whatever we are writing now and even though we are writing it as an RNN, so actually the architectures that go into building the RNN there is something that is very very popular and what is called to say LSTM. And it is called long short term memory, it stands for long short term memory. So, it looks like you have a long and then you have a short term memory, so we will just get afraid look at that and the idea is that right idea is that and there are variants of this by the way right. So, the first time that when it came out of Mennonot was more it was called long short term memory, but then there are various different versions of this and then there are other things that are that are that have been derived from this kind of thing and so on. And then the idea behind this long short term short term memory is to actually handle this gradient problem. So, the idea is that right some things, so here is where right here is where incidentally your sigmoid right begins to play an important role, I mean I think I had made the statement right somewhere in the beginning right I had said that sigmoid there are places where it is important otherwise we said that ReLU and all this what has taken over right.

So, this is called a gating, so the LSTM uses what is called a gating idea and this gating idea is like you know it is like a gate right whether you should let somebody in or not, but it is not binary right I mean it could be like you know you can say that with a certain weight I will kind of say let this person in. So, it is like gate right, so it has actually three gates, so this LSTM and the idea is that idea is that you know the way you kind of you pass information you have you kind of create a gradient highway. So, for example, right so this long term is something right that can that can take on information and that it can actually last for a long time because you actually create a gradient highway through right through which that you know information can go all the way. And then so basically that is called a cell state and then you have your normal, so the things right that you should kind of keep in mind is that something called a cell state and this is also about right I mean as I said right this is also about this one a memory only this is also about to say remembering, but cell state is something that will remember things even from the far from let us say you know from something which is far away in the past I mean that is its capability.

Then a hidden state right which is what you have already seen which is the hidden state and hidden state typically have a shorter memory I mean right because you know they cannot travel too far away in the past because there is no explicit highway for them. So, this architecture itself is created that way right, so it is actually you know it is a kind of a design thing right it is actually it is a design that allows information you know to last longer if it must last longer, but then what should last longer that you and I cannot tell and that is something that a network should figure out. We cannot tell that so you know sentiment is very important therefore I take it when it is needed or for example I throw away things right which are not which are not relevant for example I might say is are the there I mean these are all probably not even relevant and why carry all that baggage right it is probably not even relevant and again it hidden state itself how much of the hidden state really needs to pass on right. So, all three information right so kind of right there is something called a forget gate. So, this is the forget gate, gate is about is about a cell state right which is you know which is kind of see carrying information and some of the information which is out there which is totally irrelevant and you want to forget them along the way and you do not want to carry unnecessary stuff.

Then the second one is like an is like an input gate the input gate is about is about are there very very is there some very very important information from the input you know that that kind of that that we should take forward and that is the job of the input gate. So, it is like you know erase and then read right. So, the input gate is something reading from the input and then making sure that for example if it is a sentiment right then maybe right you want to you want to make sure that the that the that the sentiment is not is absolutely taken inside and then you have an output gate right which is like which is like writing right and this output gate is something like you know something like saying that you know what part of the hidden state should be exposed right. It is not like the entire hidden state needs to go maybe maybe a part of it right this is all those all is all that is needed. Now, what is needed and what is not needed? See the you see the nicety right about these about these deep networks is that right in a sense that the nice thing about them is that they are all they are all learnable right they are all trainable.

Thankfully right we do not have to sit and tell what you should learn right because if you knew that then then right it would have been an easier problem. But in most cases we do not even know when you throw in so many sentences right then the structure and then the right you know a dependence of one on the other and then the importance of some particular word out there all that right it has to it has to come on its own. For example right when when when let us say that somebody says a movie sucks and after that you write flowery words, but then you know that the ground truth is that the rating for this will be low correct, but then that it became low is because there was a damaging thing somewhere right which caused it. This is something that that you and I know, but then a network has to know that right network has to learn that inherently it has to what do you say right it has to it has unravel this somehow right when it learns and that is where the learning strategy and all becomes important. It is not just true that you throw in then then you know you just get to sit back and relax right you know let me run it on a on a big GPU system and hopefully it will do everything not true ok.

So also not a lot depends upon even the kind of data set you make whether the whether the data set is impressive enough if the data set is weak what can it learn right you can only learn learn learn things you know that you can teach it right in a way and we are still far from the days wherein you can do something in a self-supervised way ok we are very far away from there. For example we have VQA right which is which is something right similar to the one which which I which I which I showed you know in the last example the question answering a visual sort of a question answering so there right I mean a lot depends upon how tricky are you asking things for example right I mean if you say if you say right what is what is the guy what is that boy wearing that is maybe easy right you can just put down the color but suppose I say what is the right of the boy right now you have to sort of figure out right that you are you are talking about a directional sense so all that you know it won't just acquire on its own right we have to give it examples of that time right ok now I mean right before we lose time on this right so so the LST so having having said right I mean what what is what each of these gates do so I will just I will just you know I will just draw this initial picture right that if you had a regular sort of an RNN right what you would have is something like this you get an input Xt - 1 right when you know no I think I can go fast on this because I have already explained to you what it is so you get like HT - 1 and then out of this HT - 1 you might just pass it on you know act something on this in order to in order to write you know you want an output somewhere you want to tap an output here well I mean ok so you can actually well you can actually draw this from here and then you can tap an output then right you could again send it you know in a sort of recurrent fashion then this HT - 1 goes inside and then right and then the way to draw it right is like this this is a standard way to draw this ok so you have Xt right going in and then you have this guy going in and then the two will go together to to a tanh ok tanh I mean like I told you right that F if you if you if you could have right recollect for the for the for HT there was an F right and for OT I think we had a G right and G is typically a softmax and F is typically a tanh ok. So and then and then again right this this goes and and then again right again you have another and then this will go inside the same thing ok will again repeat then you will have a tanh here and then you will have an input coming from here if it is Xt + 1 then this is this HT - 1 this is HT right and then this is again tanh and out goes right and then and then all of this kind of it goes on. Now the thing is right so when you when you talk about and as I said right this has a vanishing sort of a gradient problem ok we do so this architecture is not even though it is how we started we started sort of talking about RNN but this is not vanishing gradient issue right is there. So in order to sort of in order to address that right so I will talk about I will kind of build this LSTM one and one gate at a time ok.

So the way it looks so the LSTM right looks like this. So first right we will talk about the forget gate so the cell state right like I told you is the CT - 1 that is at cell state ok typically this represent as a CT and this represent as HT yeah this comes in and then you have HT - 1 that is again coming in from the past which is of course carrying a summary just like just like old times ok and then this is XT ok and the way and the way it you know people typically write is you know so they will kind of do a concatenation ok it is all the same I mean whether you do U times what was it I mean I think we use U and XT right we said U XT + W HT - 1 that is what we are using right. So instead of that you can write something like WF and then you can have a concatenated what is this HT - 1 and XT right they are both they are all equivalent and then what happens is right this is given to a sigmoidal neural layer by which you mean that a bunch of neurons ok. So you know that so you know that ok now what comes here right is actually this one ok so this is like what to say ok now let me just write this right so suppose I call this as FT ok so it is a forget gate that is why we use F for that and FT right will have a form that looks like sigma of WF of HT - 1 and XT concatenated + whatever you know + let us say a bias BF ok. So what this actually means is that your HT - 1 and XT right after they are acted upon by a WF right which is somewhere here right that is where it is acting after that and of course right and you have now what you are doing is you know you are actually passing it through us through a through a bunch of neurons right which have a sigmoidal these are all sigmoidal neurons and you are trying to weight each one of them and you are doing an element wise multiplication here because your CT - 1 right I mean of course you have to make sure that the sizes and all match that and all you know will get taken care of and you actually multiply the two ok at this point and the idea is that is something from CT - 1 is actually irrelevant right after having seen HT - 1 and XT you will find that certain things can be just thrown out you just you just remove them ok.

Now this is just the architecture I mean we cannot pin it down to say that how did is get removed or the get removed it is not at that level at a conceptual level that is what you want, but at an implementation level this is what happens right. So, this is called a Fergate gate it is called a Fergate gate right. So, in a sense right so what you want to say is which part of so which part of or which part or parts of CT - 1 should be forgotten because they are kind of irrelevant forgotten and of course you know and this kind of goes forward ok this we will use, but at this at this level of the Fergate gate right this is how it looks. Then when you come to the input gate right now I will show the CT - 1 is dotted and then right you had this and then you know here I will show all of this is dotted because this you have already seen this is FT and then you have a sigmoidal neural layer then you had HT - 1 and XT concatenated and what happens ok this will also we will show as dotted. Now this again this concatenated thing right is actually passed through of course right normally what we would have done is you would have had had a tan H just as we saw there right we would have just passed it through a tan H and then get us taken it out, but what we now do is we actually pull out what is called CT tilde we call that CT tilde and then we have another sigmoidal layer here which whose output right after you multiply ok.

So, again I mean so at each of these places there is a matrix and all of weights ok that is multiplying and then this is what is this ok the standard notations ok these are all standard notations ok typically and this is like IT and these two are multiplied again. So, in a sense right this is starting to wait what in the input right should you know is actually important and should be taken forward ok. So, the entire thing about managing you know information is all about gating ok. So, each is a gate. So, you are trying to gate and then find out what should be let in what should not be let in what should be partially let in and things like that right and then what happens is this and this ok are then added here and the equation right here will look like this IT which is here is sigma of some Wi acting on HT - 1 XT + Bi.

So, the i and all has to so the i goes for that you know for that input gate ok and then C tilde t which is which is here ok this is C tilde t this tan H WC. So, all these will have a respective weight matrix ok WC then again HT - 1 and you see that HT - 1 XT will go as input to each of these gates and another thing to notice is that they are not sequential in nature it is not like one thing happens and then the next thing happens ok they are all interacting ok they are all acting at the same time together in order to get us to decide what should get in what should not get in or what should you know how much significance you should give to a particular thing and so on. HT - 1 XT + BC and then your final C t ok now what is C t is this guy this is the cell state C t that is C t - 1 F t right that is what you get here now C t - 1 into F t and when I say into that is an element wise multiplication it is not a it is not a this one a dot product or something ok. So, it is like element wise multiplication + this guy IT times C t tilde C t tilde. So, these are all element when I say dot it is element wise element wise multiplication ok and this is this is the input gate.

So, the input gate is here by the way right. So, input gate. So, the Fargate gate was here and then finally, let it you have what is called the output gate which should tell which part of the hidden state right should actually be taken forward. So, here again right. So, the HT - 1 and all will go common ok HT - 1 then XT will come along and then you will actually concatenate the two then you have this guy sigma the Fargate guy and then C t -

1 coming from here then multiplication going forward an addition here then this goes forward and then you have to get the input gate and then this is tan H and then all this is exactly the same and then here right this is your C t.

So, what is done is. So, you have a tan H. So, the C t is passed through a tan H and then you have again something that actually weights that and that comes from here. So, this guy will again have a sigma. So, your third gate right which is the output gate will then find out the significance of this of the output here that is it is like tan H of C t right.

So, tan H of C t how much of that is actually significant because instead of passing the whole tan H of C t it will again evaluate what needs to be passed and that becomes your HT. So, this HT - 1 and all that we are keeping on showing right that is actually coming from coming as follows through the output gate ok and this guy is its job is to determines there are many different ways to write it, but what part of this what determines what should be the next hidden state ok. And then this the same network right I mean you can just have you know in a in a recurrent manner and there are some variations of this and anyway right. So, if I asked you how many unknowns and all that you should be able to find out ok. I mean for example, if in the exam right I asked you that an LSTM right if it has whatever input of a certain size then the you know cell state hidden state if they are of a certain size then you know what will be the what will be the total number of unknowns and all that and all I think now that you know to do it for MLP and all and CNN and all and then you should be able to write do it do it for this also.

And yeah so, this is the gradient highway that I was talking about is here ok this is this is that this is that is a gradient highway ok. So, what this actually means is that so, what this means is that when you when you do a back prop right it does not it does not encounter encounter things along the way which can which can actually suppress a gradient the only thing that it encounters is actually FT and and that by itself it does not cause any issues. Whereas this HT and all right has to go through go through you know several other other what you call other paths because of which right its gradient does not survive for too long that is the reason why you call this is a long short term memory the long term memory coming from the cell state the short term memory coming from if coming from HTs and then a gating mechanism that actually that helps figure out automatically as to what is actually relevant and what is not and that that in a sense. So, whenever somebody says RNN or something right they will typically the implementation you should think that probably is happening in terms of an LSTM or to what are called GRUs gated recurrent units I mean which are which are slight modifications of you know which are all again based upon this fundamental LSTM and you can actually show that you know under certain conditions you can ask this question you know can I can I what should I do to these gates right in order to make them equivalent to an RNN is that possible those are all are things that I will leave it to you and you can just think about under what conditions right is there something that I can do so that it looks like an RNN right it actually behaves like when I say RNN I mean the I mean the original RNN right that we had which was a simple structure tanh followed by softmax or something right. So, I think those I think I know I will probably skip and yeah and I think you know yeah right this is all this is all I wanted to talk about in terms of the review part ok and starting right next class which is today is Wednesday right so Friday right.

So, we will start the other thing which is a computer vision part and the goal would be to kind of go something like you know we will do initially is initially features because you know just like you saw features and all here there are actually you know handcrafted features that are very very useful and even today are being used. So, we will go through features first then we will see how those features can be used to solve you know geometry problems and then and then kind of go on from geometry to mid level and then high level vision so I think yeah that is that is the path we will take ok. So, I will stop here for today.