Modern Computer Vision

Prof. A.N. Rajagopalan

Department of Electrical Engineering

IIT Madras

Lecture-29

Okay, now today right I thought we will do recurrent neural networks, okay recurrent, these are a little away from the ones where the, I mean of course they are also based upon same principles as we have already seen but there are certain other things that get involved when you talk about recurrent neural networks, some things that you have not seen before and it is mainly used in language processing, okay mainly but does not mean that you know it is limited only to that. Whenever you know there is a sort of a history right which you have to remember there is a memory involved right, you need this kind of an architecture, okay just that the architecture right has a name and it is called recurrent neural network and yeah, so whenever right I mean for example think about, think about you know predicting a word right, suppose I gave you a bunch of words I mean this you often do right when you are kind of typing in you know there is an anticipation as to what might be the next word right that comes because of the history right and it is not just the previous word, there is something before that that probably happened and something before that that probably happened right because of which it is able to predict with certain probability that maybe the next word that should come in as this and then we take it or leave it right, I mean when we, but the training right is such that it has seen lots of sentences right and something like Wikipedia or something right from where you can take lots of sentences and then train right and so that a network knows to anticipate given that the past history was this. Again you know there are lots of things okay and this is a fairly deep topic you know in the sense that how far past you know how far into the past should one go right, where are all the important stuff because sometimes right something might have been said right in the beginning. For example, if you think about sentiment of some movie or something right and maybe so maybe something very damaging was said right in the beginning and then right if you miss that right then you actually miss summarizing right what it is probably about. So I mean it is a little tricky okay but I thought here I just wanted to give you a flavor about what it is like right and you know and just I read if at some point of time right if you had to do I do not think we give you any assignment on RNN right we do not give you but then it is good to know as to read how these things work okay. So in general right it is called a sequence to sequence learning okay, sequence to S2S to sequence learning and it is because of the sequencing that you have a temporal ordering and then a temporal sort of a dependence and all that right.

So sometimes they call it S2S right sequence to sequence learning. Some examples right I will give you just to just to begin with right let us just take a few examples of what where you might encounter these. For example one could be language translation right language translation like I said right I mean many extra applications involve text but does not mean that it is only text for example image captioning you know or you know where for example the output is actually is actually a text but the input is an image right. So it is not like every time the input and the output both have to be text but yeah but in many cases right you will have text coming in.

So for example let us suppose I say hello how are you right okay if this goes as input right and then what you expect from the output let us say it is in Hindi then maybe right I might say Namaste aapke se ho or something right Namaste aapke se ho or something like that right ke se hai whatever right aapke se hai right. Well I have forgotten my Hindi and all but something like that right. So that is like a language translation so and so the input is actually right is an English sentence and the output is something like that right Namaste aapke se ho and then you can have case like speech to text right this again 0 things that you see around you right there is nothing unusual speech signal to text. So where let us say right when I get a speech waveform right that comes as input and then maybe right it means something maybe right it just means hello how are you but then I want the output to be a text right. Input goes as a waveform and then there should be a network and it should kind of you know analyze this waveform and give out an output that can be in a text form.

Then there can even be what is called a question answering see this is again something you know that is very common you know a question answering. So for example you know I mean yeah it depends I mean there is something called a visual sort of a question answering what it means is that I give you an image and then I ask something about that image right that is I give you an image and then say what is you know what is a boy actually playing with your answer would be probably a one word answer on a ball or something or you might say what is the color of the dress the boy is wearing you might say whatever right. It can even be sometimes a sentence but typically visual question answering is like a one word answer you know the output I mean a question goes in along with an image that is like you know a visual question answering. So you can also have a simple sort of a question answering but more common is what is called VQA okay a visual sort of a question answering. Then you can have like I said an image captioning okay and another example could be like image captioning and that is also the reason why if you look at the different different examples that I am giving there is also the reason why actually you have got different flavors of RNN okay depending upon you know whether it is single output multiple output single input multiple output or multiple input multiple output right depending upon that that you can have different flavors okay.

Image captioning right this is another example. So in this case you just we just push in an image and then out comes a text right. So output is actually a sequence or a text typically caption means it has to be some kind of a sentence which we don't know a bird flying over a body of water or something like or the or the or whatever it is right I mean you want to say you want to summarize right what they see in it or a or a boy is playing in the you know in the ground something like that right. So and again that all of this will require some kind of training right I mean it won't automatically happen it is not like a human being right where we just look at an image and then we make interpretations about what is going on right. So again all this will require that is the reason why what problem you attempt very much depends upon what is available out there right.

If somebody has created a data set where you know somebody you know kind of sat down and did a painstaking work of you know giving an image producing a text corresponding to that and saying that that should be the caption that should ideally come out okay and all that. But one of the things which you will also notice as compared to right things that we have seen earlier is that now you even need something like scores for you know a text right I mean you are saying that a boy is playing in the ground. So you need to understand I mean how well is that is the sentence construction I mean would that have become bad I mean if I had interplayed the words or you know inter placed in the words right in some other form right that might actually change the meaning of what is coming out and so on right. So therefore there is a lot more to it than just the simplified version which I am talking about okay. So captioning is like you know image going out and then the output comes out as a sequence then you can have time series data which is very you know which is very common which you could write in the beginning anticipated time series data and so on okay.

So one can think about so many places right where okay where let us say you know where you have this kind of framework right and it is not clear right for example at this point of time if I asked you right I mean can you just use a CNN or something or an MLP right that you have seen before and could you have just solved all these problems with that then the answer from you would be like not sure right because it is not clear as to how do you sort of what do you think is really strikingly sort of you know this one a different in these set of problems right the most of the ones that I am talked about here compared to what you have seen before. Till now we saw no we did CNN we did MLP right we had a certain kind of input data going in we were doing some training that was a loss function that we had at the output just at a top level right if you were to think about what is probably happening here and why maybe such an architecture may not be may not be of use or not cannot be directly you know employed right for a task like this what would be the things that would catch your attention? Frequency also a little bit. Correct yeah so in a way so

what you would call is there something like just like we had things like what he said it before you know we had things like you know what you call dependencies and so on at local dependency and all we talked about right so is there something similar what you said is correct but then is there a technical term that you can put in there that sort of encapsulates right what you are saying a temporal this one a dependency right there is a temporal this one a dependency which you did not see till now right I mean in all the things that we did there was never a temporal dependency of course you know if you had taken videos or something maybe right at that point at that time you would have seen some kind of a that is why in fact RNNs are useful for that kind of you know temporal this one dependency. Okay so one of the striking things is that you see already a temporal this one a dependency and what about others temporal dependency is one thing that is plain obvious that is going on here which you know we did not see before anything else? That will all have to be learned by the network right we are not going to specify anything explicitly right I mean you could just take something like Wikipedia or something and take sentences from there and keep training. Another thing is that I think that you should definitely you cannot miss this the fact that the input and the output sizes or lengths what is that or not fixed exactly input and the output sizes or lengths or dimensions whatever you want to call it sizes vary from example to example or not fixed in the sense that they vary from say example to example.

See for example sometimes you might have a long sentence for which you have a translation or something sometimes I might have a short sentence again I need a translation for that and there is nothing like a fixed dimension right the input can change its dimension the output from example to example during training output sizes vary from example to example or from one example to another. Okay so in a sense these are kind of two striking things that are happening now which we did not see earlier. Now there is just one thing right which I should probably mention that is that you know when like I said right RNNs are mostly used for language processing because in languages there is inherently a temporal descent dependency even though videos and all also have it but mostly you know languages is where you know you have to capture a lot and that is why I read you need a representation. For example when I give you a word right I mean I need some kind of a representation I mean I cannot just throw a word into a network right how do I how do I even input a word right. So there should be a representation right now that you know representation is typically done using what is called a word to wake.

Okay this you will see I mean or there are even other representations but something word to wake what it means is you know you convert a word to this one a vector. Vector means it is like a set of numbers you see finally you can only train with numbers right you know you cannot do anything else you know network has to understand how to read numbers. Now what is done is I mean that we are not going to go into how this word to wake is

formulated and so on but the idea behind word to wake is that right when these words occur right there is always an interplay right between the words when some word occurs there is typically a bunch of words that occur right around it. If you go through text it really watch very closely all this has to be of course captured we cannot manually sit and do this so this word to wake is actually something like that I mean it is actually a network by itself which has been trained over lots of data and where it understands the relations between words right when something occurs. So it is like saying it in a space okay if I have a feature representation for a word then I need a representation for another word if that often occurs when this guy efforts then I would need something which is kind of close to this representation right because these two sort of you know seem to you know it looks like when this occurs that guy inherently also occurs or you know occurs you know let us say two words before three word depending upon how far away right I mean did you follow that.

So it is like you know it cannot be like you know one is here and one is way elsewhere but then actually you know they do have some kind of closeness when they actually appear in actually sentences right. So that is what this word to wake actually captures so it is a set of real numbers okay it is not like a one hot because one hot represents completely miss out this space this kind of a relationship I mean you might ask why not I have a one hot vector you know the I will have 1 0 0 0 dog maybe I will have 0 1 0 if you do like that right then this you do not capture this kind of a relationship that actually inherently exists okay because words do not occur at random right sentences are constructed based upon something right based upon certain rules and therefore they have a spatial or they have a relationship in that sort of in some sort of you know feature space and that is what this word to wake captures okay. So from now on whenever I write a text let us say I say the boy you know the a boy right so boy will be like a word to wake right so you should understand that is there are bunch of numbers that are going it and that actually means that those numbers actually have a representation that means that you know that is a very nice representation it is not just a one hot vector it is a nice representation that actually encapsulates the what all so if you look at the vector representation of words around boy you will see that oh maybe when boy occurs ball probably occurs and that is for ball and boy right if you see that vector representation there will be somewhat similar otherwise you will just you know throw away all this relationship and build something which is you know which is not very meaningful okay. So this word to wake is something okay that we will always remember that you know we can do there are other representations also but just keep it simple okay it is called word embedding okay I am not going to through that okay so then right so in a general way right if you talk about RNN then it looks like what can go as input right is X1 with all these examples okay X2 and then XT okay XN - 1 XN and then output will be can be like Y1 Y2 and then YT okay and then and so on okay. Now see one way to one way to write think about think about this is that I mean why cannot

I have for suppose let us see one of the simplest ways right I mean in which right you might be able to handle probably this variable length and all that right what you could say not variable length let us say one way to one way to let us say right in a very naive way right if I had to do an ML do this RNN what I could do is the following right so I take XT - 1 which is my input okay then I construct something called HT - 1 right I mean I will talk about what is HT - 1 is then I have something like okay now from here to here I transform it using a matrix let us say UD - 1 then I have OT - 1 which is my final output and then this I transform this HT - 1 using some matrix VT - 1 and maybe there is a bias and so on and this is this is actually an MLP okay the MLP right which you already saw so this so this so this HT - 1 is itself like you know U times you know UT - 1 times XT - 1 so there are bunch of weights out there and there is a bias out there and then maybe right there is some there is a no there is a there is an activation sitting out there and therefore right you get your you get your HT - 1 like that and then on top of that right you can have your VT - 1 is another thing that is acting on top of that and then you have the output layer which gives you your OT - 1 right so this is what is what we understand as an MLP right now then I take I take the right next input which is like you know XT and then for which right I can have a matrix UT and then right I kind of produce an output OT and then VT okay there will be another another matrix and then maybe you know next next instant of okay this HT then the next instant of time I get an input XT + 1 and then I multiply it with UT + 1 then I have VT + 1 and then I have an output OT + 1 I can do this but then it when one of the things that you did you clearly notice is that this kind of a temporal dependency right it is not it is not able to able to encapsulate right it is not embedding that right I am not able to see that something that occurred in the past is being used in order to make a prediction about about even next next output for example OT that I do not see how let us say OT right OT it looks like you know whatever XT comes in it is just acting on XT and then doing something it is not even it is not taking the history into account right that is one thing and the other thing that is that is also looking a little jarring is that these weights right tend to will tend to blow up because you know every time that you have UT - 1 then you have a you know different matrix which is UT all these are weights right finally then you have VT + 1 UT + 1 so this should be UT + 1 ok and so on so your so your parameters definitely do not look like you know and the other thing is that how many MLPs do you need right I mean so each is a I mean so each is an MLP now and it looks like you know I will have a varying length right sometimes I may have to use more sometimes I might have to use because my my input length can be actually varying right I mean here of course in here I have a kind of a kind of a synchronous thing it looks looks looks like I am asking for every input right I am asking for an output sometimes it can happen that I watch the entire thing and then and then give give only one output right and some like action recognition right so I will watch the whole video I mean I cannot just just put out an action for every frame right action recognition is something like you know I have a bunch of I have a video so I have to go through all the frames look at what is happening and then

finally output output something I will say it is whatever it is a max running jogging something right so so here so so that is why it RNN can have different flavor it does not mean that every time you will have a setup like this but the simplest which you can think about is this right for every every input that is coming in maybe right you are asking for an output okay but but then that clearly okay it does not encapsulate this and the other thing so so right the weight seem to be a problem the the the I mean the variation in the science seems to in the size seems to be a problem and third is a dependency right it is not it is not being encapsulated so there are various versions of this I would not go through them but eventually right what kind of one sort of you know one settles down for right is this is this structure okay which is like this which is a okay so you have xt - 1 again going as input then you have u so the error dependence on time is gone then you have ht - 1 okay ht - 1 okay ht - 1 and then you have a v and then you have ot - 1 and there is a connection from here to the next this one which is ht and this takes xt xt and this is again u this is again v this is ot and then again you go like that then you have ht + 1 then again the same u then the next input xt + 1 ot + 1 and so on so now what has happened right most of the things that we said are kind of handled here one is the I mean one is the weights right so which part have we used here which property right have we used here in order to reduce the number of weights we are just weight sharing across time right we are just using the same bunch of weights we are not changing the weights right so it is like weight sharing that we started we talked about when we did a convolutional neural network something similar to that we do not change our weights we are just sharing the weights across time now we just keep them steady constant oh sorry and there is a w here okay there is a w here that is also a matrix of weights and the other thing right that we are also that we also seem to be doing is that okay now right I mean in a way right what you should kind of you know look at this look at this effectively right given the okay now the other thing is a temporal this one a dependency right I mean that you can see is happening because of the fact that right this one okay this quantity that you have here right I mean it is called actually a hidden state or it is called the state and this kind of it encapsulates whatever happened it can it is like a summary of whatever has happened till that point of time see for example right I mean if you look at HT right HT seems to depend upon HT - 1 and then it seems to depend upon XT right but HT - 1 in turn right depends upon XT - 1 as well as HT - 2 and HT - 2 in turn in turn right will be a function of HT - 3 and XT - 2 right so in a sense if you just look at the if you unroll it right if you unroll in time you will realize that till whatever point you are if you look at this state vector and which is like you know HT - 1 or HT for that matter that has encapsulated all the information coming from the past it is supposed right we wanted to encapsulate all that information from coming from the past and take this take this new information right which is your new XT that is coming in and in a sense right make an estimate now and you know this is not anything very new okay by the way if you have done a Kalman filter or something right it does exactly that right have you done a Kalman filter where you have a state equation you know which talks

about how the how the variable that evolves over time you know as a function of itself so for example you can think about a smooth motion right of some object now that could be a motion model right that you have so that even if I do not observe anything for some time right this happens when you are doing target tracking a target goes right I mean you know there is a cloud which kind of obstructs your view of the target right so there is no observation at that time you do not see it at all but then you will have a motion model which will say that even if I or if I have a very weak observation I will still go strongly by what my motion model says because this guy was moving therefore you do not you do not expect it to suddenly suddenly stop somewhere break off if it breaks off it will go wrong but typically that motion model will actually predict so this this kind of a prediction right that is happening is coming through this HT so the HT is sort of saying that all the past right history right so I mean you know in terms of tracking right it will be it will be the it will be the motion history of that particular point or something whereas here typically it is for languages right so it is like this like the history of the sentence or something or it could be a video where the history of the temporal action the video is evolving and I mean right if you have done if you have done the see HMMs and HMMs right I mean what are called hidden Markov models they have all all used these notions of a hidden state and all that okay so this is so the hidden state is something that is supposed to encapsulate information coming from the past and then then you have to use the current input and then make this one you know a prediction okay so so that way you have kind of you have been able to take care of the weights right you do not have a blow up and in a way right this a succinct representation of this right is this so you can think about this as H and then W and then X then U then V and then O so in a sense right what is happening is you are just using the same MLP again and again it as as this as this input input keeps occurring that is what is going on right the unfolded thing this one unfolded in time is what I have shown on the top that is like unfolding it over time but really what you have done is you are just reusing the same MLP right as one input comes in other so so that way right you no longer right I mean you know you will know you no longer have an issue about issue about right issue about you see varying lengths see by which right by varying lengths again right what you have to kind of see realizes that I can have a sentence that is that that consists of 5 words let us say for which I need an output the output again can be 6 words 10 words we do not know whatever it is right I mean but the input the next time right I might have an input but has probably read 8 words in it that is a another sentence right so now you do not you do not need to a priori know what should be the length because right that was a problem that we saw initially I mean how many how many should I put there but now because of this because this is rolled up sort of an architecture which is why it is called a recurrent recurrent structure right because it is recurrently acting and therefore because of this recurrent nature what will happen is we do not have to worry right if an input is 5 words then this recurrence will happen 5 times sorry not word input sentence even for a word you can have right if it are counted in terms of the alphabets and so on or if you have a

larger sentence it will automatically right act right that many times and so on you right you see the point ok.

So so in a sense right so in a sense so the RNNs have actually memory ok that is something that we never saw before have memory and they are called recurrent as they perform the same task called recurrent as they perform the same task for every element in the sequence for every element in the sequence every element in the sequence. So, it is like a chain of repeating modules right.