Modern Computer Vision

Prof. A.N. Rajagopalan

Department of Electrical Engineering

IIT Madras

Lecture-31

Okay, so today right we will just do this part which is on encoder decoder models. In RNN, these are important because most of the applications right in RNN come under this this category what are called encoder decoder models. And the idea is that right you have, so when you say encoder decoder what you really mean is that is that you have you watch for your entire input. The input could be you know it can be whatever it can be text, it can be image, it can be text and image whatever it is right and that you watch you know completely and then have a sort of representation which is an internal feature representation for that which you then get as you pass on to a decoder. The decoder in turn can be an RNN again or the decoder can be a simple MLP depending upon the final task on hand. And these two together right act in a manner that depending upon what task you have right it can be solved and that is why these are called encoder decoder models.

And many of the applications that have emerged within the realm of RNN actually fall under this. So the best way to understand this to actually take an example. So what I thought is I will take an example just to illustrate the point. So this is one example of an encoder decoder model.

So which is an example case is image captioning. Okay we have of course referred to it multiple times earlier but now we will watch it in a sort of a more careful manner. And the idea would be that right tomorrow if somebody gives you a different problem right you should automatically be able to figure out how that encoder decoder how will it fit I mean if it fits into this kind of a right this kind of a kind of an architecture then how will that architecture be right that is something that I think should be easy for you to for you to be able to do. Okay so we will take an example. So image captioning.

So as we know right what goes as input is actually an image and then what should come out is actually a caption. So which is like a sentence right that should come out. So as far as the image is concerned so what will be the decoder and what will be a decoder here. See I have to I have to do image captioning right. So I have an image and I want I want a caption for it.

All this of course has to be trained and all that right. I mean we are not worried about that

training at all will happen let's say. But what will be the encoder's job in this case. It should take the image and then actually write no and from the image it should be able to draw out you know a visual feature out of that image right. And like I said you know generally you don't just pass the image typically it will be some you know feature that kind of characterizes the image depending upon it could be like FC7 layer in an AlexNet whatever it is.

And the decoder in this case you want a sentence to come out. So the appropriate architecture right. So for the input for the encoder part it looks like a CNN is appropriate right. You don't need an RNN for that. You have an image and you want to kind of extract you know features out of it which will be a visual features and those features have to be sent to you know a decoder and the decoder's job is to actually look at this visual look at these visual features as a whole and sort of come out with a sentence which of course it will be trained to do that right.

Can't do it on its own and that output right you will you will you will of course you know tend to you know agree that that has to be an RNN right because it has to produce word by word it has to go and produce a sentence right. Similarly if you if I give you another problem like for example suppose I told you that you have to do action recognition right given a video. Let's say there are multiple actions that are there and I want to produce I mean I want to do that problem what will be the encoder decoder model. Suppose you have to do activity recognition right. I mean all this many of these will come under this kind of a framework an encoder and then decoder.

So what will be the encoder what will be the encoder's job I mean what kind of an architecture do you suspect you will need for the encoder now. See activity means what it has to be some kind of a video. From one image it's typically very hard to tell any activity. Typically activity means there is something that you are watching okay and it's typically and typically a video. So that's your that's the input that you have and at the output rate you are asking for what is that activity and it is somebody drinking or somebody whatever it is sleeping typically there are some you know each data set will have some actions that it wants to classify.

So what so what do you think will be the encoder. Encoder will be an RNN right because it has to watch frame by frame frame by frame and then it has to summarize all of that right. I mean it has to watch all of that and then then it come up come up with some you see summary feature which will then be passed on because you can't make out an activity but is looking at one frame and you have to watch the whole video and then what will be the decoder now. Decoder will be a simple MLP right because all the data should do a classification right I mean you have some n number of classes it needs to tell which of

those actions it is right. So again so worried where the RNN will come in or how it will come in right these all a function of what problem we want to solve okay.

So in this case clearly you have an image that is going in okay therefore it I will just write down I will just draw the architecture and you know it will be obvious that that's how it ought to be right so you have an image right that you have as input then you look ahead of push it through a CNN and then out comes a feature let us call this as S0. This S0 could be something like an FC7 okay of that image whatever right some feature representation then this sort of a summary feature will then go and the and then the H0 right which is the initial state of your RNN which is supposed to produce in this case what it has to produce a sentence right. So it will take this input which is the image input and then the other input that will come is I think the other day somebody was asking I think he was asking right so what is called a token so you have to say go right that means you start now right start kind of right throwing out the words you know so it needs to get a spit out one word after another so typically right this will be a one so this will be like go and this is a token that is understood it is actually a part of your what is called a vocabulary right. So you have this one vocabulary of words in which stop and go or start whatever you want to say right that will be that will also be a part of the this one so this we can say it is a kind of see it is a kind of special word right it is a special word in the vocabulary okay and suppose let us say suppose let us say now the caption is such that let us say something like the cat sat on the bench or something right the cat is sitting on the bench or something cat sat on the bench okay if this is the caption right that is supposed to be output then what it means is right that this guy takes it and all those things will happen right in terms in the sense that this S0 will have to be acted upon by some by some weight vector weight matrix and then this input right will then have to be operated by some weight matrix here U, W whatever we said right. Of course sometimes that people differentiate the W's that come later for example after this also right you are going to have this going to H1 and so on right now that is what we refer to as W so this they will say as the one that is coming in here right this will be called a Wconf because that is a weight matrix coming from a convolution kind of an output right it is not CNN whereas these are all part of the RNN so the W so they do not have to be the same right so we will typically say WW for the rest and for the initial one you might say there is a Wcon or something and right so the first one could be right could be could be like let us say probability of the right that is what that is what is my that is what is my first word right that I want and the second input right that that needs to go here right will be will be typically the word see ideally you would want the to come out but but right we do not know we do not know for certain whether it is going to it is going to throw out you know the word the but whatever is this word so so at the output right you have a cross entropy loss now right then because because you you are kind of right expecting a certain sort of you know one heart this one representation to come out and the input that goes in right here is not something that is going to come from elsewhere

because I have only that image right so so so this input right in a sense will come from here whatever was that word right that was spit out its word2vec representation right so that is what I said last time right there is always a word2vec representation so that word2vec going to say representation will go as the as the you know as the input for the I mean next state and the next state will of course you know draw from H0 also because H0 has a history of the image right I mean this has a visual feature so all these so here are your visual features right so that you have extracted from the image so your visual features could be in terms of you know whatever there is a cat laying there it is all that all that has to be going to see capture right within that visual feature and this H1 right will actually take whatever was the whatever was the previous output and then and then it is supposed to spit out let us say cat right so it will say cat given whatever it given the pass output and IF and then that will in turn right go and then then you will have the H2 and then and then you will have probably cats at whatever right given that whatever whatever it occurred earlier and so on and that way it will go until you hit a stop stop token at which time it will stop it is something like this so so here right you see that this part is the encoder part right that is the encoder part and this part is actually a decoder part and in this case the encoder happens to be a CNN and the decoder happens to be an RNN and in terms of the actual actual equations right if you look at it we have to get a look at a loss now let us sayθ and thisθ of course other than the bias rate what you will have is a W okay so it will be all this the set of these matrices one is Wconf which is from here right from so from from here to here in going from here to here right I mean you need a matrix that will then take that and right and send it to H0 then you will have the UVW that are normally there sometimes you write them explicitly as UD, WD, VD just to represent that they are actually decoder they are coming from the you know decoder side so you might say the UD, WD, VD just to emphasize the fact that these are all on the decoder side because on the encoder side you do not need anything and some bias here and there that you might need okay so Lθ right if you look at it this will be like summation over whatever at equal to 1 to T the number of words and again this will go example example by example wise during training you will have different different images that you have to give for which you have the captions so when you train that is how you will train so when you say T equal to 1 to T so that is this cost right that is going to happen and as I said this is an unroll right the same network will keep on unrolling and depending upon adjust depending upon the length of the sentence it will keep adjusting itself right so we can accommodate varying sizes not a problem right it does not mean that all sentences have to be the same length and so on they do not have to be and this will keep unrolling accordingly and you will have a cross entropy loss if that is what you are looking at then you will have a cross entropy loss which you will represent as L T(θ) and this L T(θ) right in turn right what will that look like right so the idea is that right you want to be able to kind of well I mean right if you want to solve it as a minimization okay so let us see right so what you really want this probability that let us say depending upon what we represent this is let us say the output right that is going

to see coming out suppose we call that as Y hat at okay this is a T + okay at any T right suppose we represent that as Y hat of T then what you are saying is probability that Y hat of T is equal to the is equal to the true word right in a sense right because we know that what we actually want there we know that okay we know that for example here here and I know that the output should be a cat right that should that is what should get flagged so Y hat of T is true word given that given the past Y hat T - 1 and the image I F and that is what that is what is the that is what is the or in general right you can simply okay you can simply say X T or you can say I F right it does not really matter and this right in turn because of the fact that and one more right one more thing that you will find is that typically okay I typically read you will find that you know people will write H T to be RNN of H T - 1 , X T okay that means at any at any instant of time right if you are looking at the hidden state vector that is coming out of the RNN network where in the RNN itself right is really dependent upon what it actually receives as a kind of a previous this one right a state hidden state and the current input X T that is how it is not at any point of time you have a current input which in this case incidentally happens to be the previous output does not always have to be like that but in this case that is how that is how it turns out to be and then you have a previous state which is kind of sense summarizing whatever has happened before right in the way back into the past and in a sense right you can actually write this down as Y hat of T okay all this rate is equal to you know true word and all of that and the best right the succinct way to write it it is as follows so you can just write this as so H T right that is fine okay and this X T okay there is something else that I should I should not forget to mention when you say X T right this you will typically see that you know people will write this as you know E of E of okay in this case right I will write this as E of Y hat T - 1 okay what does E means it is an it is an actually embedding okay it is an embedding that is like word2vec or something it is a word embedding of in this case X T is like you know the previous output right Y hat of T - 1 so it is a word embedding of Y hat of T - 1 for this network right I am writing this in general you can write this but for this specific case X T takes up you know takes up the form that it is an embedding of the previous output previous word which is Y hat of T - 1 and and this one right so this L T of θ itself right you can write this if you are solving it as a minimization typically whenever you call say something as a loss rate you want it to be minimized right nobody says maximize or loss right I mean you may say right so that is why we will always introduce a - log or something because ideally you want to get a maximize this probability right you want to you want to maximize the probability that Y hat of T hits the correct word right that means these parameters of this network should be such that if I give if I put those parameters into this network then the probability that Y hat of T takes up the correct word right is higher let us say highest right in a sense and minimization will mean that you just take - log right that just that is just a matter of writing everything as a minimization so this further right you can just write it as Y hat of T and then Y hat of T I mean say I am kind of you know they are all the same Y hat of bracket T is the same as Y hat subscript T and all that

sometimes I think I am changing that notation and you can say that this in fact you can write this as H T , let us say in this case I will write this as F C 7 of I F in this case it does not so this can keep changing depending upon what network you use and Y hat of T - 1 in fact right because H of T is anyway capturing all of the past information so as so the so the equivalent way to write it is simply this is the same as right trying to minimize L T(θ) which is - log of and and and and write this this this itself right I mean you know P of Y of T is but softmax of what will that be the final output in terms of H T in some V H T + some bias right last time we saw no the final output rate has to come see the output output won't just won't just won't just come off here right it has to go through a V matrix right there is another V that is acting on that H T right so this is in fact softmax of V H T + some whatever bias right this is okay right that is what that is what it will be right so so this probability is equivalent to having a softmax function sitting there right so this is how this is how this is how effectively it will turn out to be and and and solving this problem is equivalently having a back propagation through time the it will take example after example after example for which the captions are already available and you can you kind of keep on training this such that at the end of the training rate you get get a decent sort of a score now there are there are certain metrics right that go with actually computing the accuracy of you know of how well words are reproduced and so on we won't actually go into that but at a simple level right this is what is how it looks now let me take one more example and then that will actually make make matters even more clear so the other one is query based the second example right that you know because this is something that you know because we are not doing auto encoders and all rights in this course so I thought at least a good idea of this will be I mean we can get a register in your head so a query based image captioning this is also image captioning but now it's kind of a query based image captioning so what do you what do you think it might be query based image captioning what do you think might be the might be the input for this so you have an image right and there is a query on that image that means let's say I want to say what is the what is the boy wearing okay that is a query so I also I also give this network an image I also give it a give it a question what what is the and then the output can be just you know one word it can be like you know what is the boy wearing or you know whatever you know what is the color of the shirt you will say brown or whatever right and so so in such a case right again this goes into some encoder decoder kind of an architectural model there what do you think the the encoder will be like no you have an image and then you also have a sentence going along with it right I mean you need both so you need an encoder that should capture features coming out of the right the image visual the visual feature as well as the textual feature it both you need okay so what would you do yeah exactly right so you need a CNN to capture the visual features coming out of the image and then you need an RNN that would actually right pull out the pull out the features coming out of the sentence and then typically what you will do is you will do a concatenation I mean that is the simplest way to kind of push the push the total feature vector inside I mean you can of course you know

think of other ways but that is the most standard way the easiest way even not just for text right even let us say right when it comes in fact this is an open problem you know in a sense that suppose you suppose you wanted to combine right video with audio right think of a situation where let us say where let us say you are you are kind of navigating from one point to another on the one hand I have I have I have a video right I see that I see my scene in front of me but then with that right I might be able to navigate to a certain extent but let us say there is an audio that I have in the sense that I have as I mean I have something that I can send out as an echo right and then you know it comes in and bounces off the various objects and then it comes back to me now if the whole idea is what I need to know what is where right in order for me to navigate I should not run and you know run into something that is an obstacle so you need some kind of a depth map for that right so that I know I navigate properly now you can ask what is the best way to combine this audio feature that is coming which is also I know this is also carrying information about the scene for example if I had just my eyes closed right this audio will still help me in some sense right that maybe something is coming from this direction from that direction I get a sense of what is happening and then vision of course that will give you information now what is the what is the best way to kind of know where should this audio come in these are all open questions and nobody knows for example you build a network with just just a vision as input you may get something now you say that I have the audio now you know at what layer right should it come right in the input should it come somewhere down the line right these are all open questions nobody but simplest way is people will just concatenate and that is the way this normally done but I but nobody can claim that that is the best way to do it and it is not easy to tell as to when it should come in and that is where people do a different different kinds of studies to figure these kind of things out but I do not think there are very good answers for that right as yet okay so this query based image right captioning will be something like that so for example I have this image right for which again I can use a CNN and this becomes let us say okay and then there is some say WCon here and I call this feature as as HI this is a this is a visual feature and and I need to I need to read the whole sentence right you not know what is being asked so what I will do is I have I have this whatever this whole sentence right what is okay in this case what I am asking is what is what is the color what is the bird's color okay something like that what is the something okay the bird's color blah blah blah birds okay the is in between what is the bird's color I mean I do not know what I saved by not writing that what is the bird's color okay so the final feature that comes out of it is some some HT it is a textual features I is a vision or image feature and what is done is this guy will go here and then that you have okay so this goes and you will kind of do a concatenation these are this is called concatenated features where did you see a concatenation before in which architecture was that Alex net dense net do you remember I mean there I had said that you would you would you would have a skip connection right going from going from every input to every other output and you go and get a say concatenate right so this concatenation

is something that will keep on encountering here of course it is a concatenation of a different kind you have audio you have a textual feature sorry you have a textual feature right this one the textual feature and then after this right you can just have a simple MLP right and then out comes the class also here right you have my class in the sense that the color or whatever right which you are talking about so you may say that right there is brown or something from a set of colors right now yeah so I think right this in a sense gives you an idea about how this RNN right is actually effective and how it can be coupled with whatever you have learned before see right so the idea why I wanted to present this was because this kind of brings all of them in under one roof MLP you learnt right so MLP anyway is also part of RNN in any case and then CNN you learnt but then again that is not really a standalone now right you can have applications that all are coming together the RNN idea is there the MLP idea is there the CNN idea is there they are all coming together in order to solve a problem okay now there is just one more thing right which is kind of the last thing right which I wanted to talk about that is that is this kind of vanishing gradient problem right of course you know I know that you know that we haven't talked about it about it in any sort of a great detail but I did mention that in an RNN right so in an RNN for you to be able to carry on information for example that I told you that told you that the history right sometimes for example if you are summarizing something like the simplest example that you can give is a movie right so when you are summarizing the review for actually a movie right then you are watching for sentiments and so on right I mean how you are saying so so so so right so let's say somebody says it starts with saying by saying that the this entire movie sucks and then after that whatever he writes right that that sucks is something right that he wanted to take in it because I mean initially it is declared that it's a it's a bad movie right but after that you know if he writes you know flowery words glorifying and all but you still don't want to accept that because you began by saying something but you see question is that information it should not vanish down the line because you know if the fellow could write a long review and that he said right something bad about the movie in the beginning would have gotten forgotten by the time by the time you are kind of ending up ending up you know reviewing the movie right so what so this vanishing gradient problem is actually a is actually a real problem that is there in all architectures and more so in RNN ok now and and it's impossible to you know one way to sort of rate you know do it is to simply say that you know you should only we should only look into a finite past don't go don't go too far into the past but but then right that's not that's not a good way to do things I mean you shouldn't be truncating because you don't know when and where things are relevant.