Modern Computer Vision

Prof. A.N. Rajagopalan

Department of Electrical Engineering

IIT Madras

Lecture-30

Now, in terms of the math right, I mean a simple way to write this is suppose I write H (t), with H(t) I can write this as a function of w H(t – 1), this again F will be some activation right, non-linear activation typically w H (t – 1) + u X(t) + maybe a bias term right. So, this is an MLP right. So, now you know what this structure might be like and then O t that is your final output that can be some g and typically right this F is actually typically a tan H kind of function ok and this g is typically a softmax ok and g of v and there are various reasons you know why let us say right people feel that tan H you know is more you know was is the most commonly used you know one thing is one thing is read tan H in terms of its stability in terms of the gradient. I told you right there is a vanishing gradient problem and there is a vanish and there is an exploding gradient problem and all that becomes even more intense here because you might need something right that was that was way back in the past right that might still be needed to actually make an assessment assessment now but but right if that if that no but no if that fades away right by the time you are you are doing a process you are processing the current input then then I mean it is not actually a good thing because then you are not using that important input that is not attention by the way attention is something else ok. So, and the other thing is that right languages like I said typically this word2vec has actually has actually got a real numbers it has both the negative values positive values and all. So, that is again another reason why they say that you know tan H is better ok.

It is just that just an just a just a thing right that has been accepted and used over and over. So, V H T + let us say another bias right B ok. This is what you have G of V H T + B and this is called this is called the hidden state ok that I think I have already written. So, this has a summary of all the information summary of all the information till time T from the past of course, and then right and then you can think about this right if my xi or xt or whatever it is of is of a dimension let us say n and this will come from where word2vec right.

Typically I mean if it is a word then it will come from word2vec if it is an image then what will happen? Let us say I write instead of word I have an image. So, if I wanted a feature well you could use them, but then normally right one does not do that right. So, just like just like you know for a word right you want you want you know a feature representation. Similarly even for images right instead of pushing the image you might have you might have a representation that is far superior than actually using the intensity values what would that be or where would you where would you go I mean if I said that use the feature representation for the image do not just push the image inside. Convolutional neural networks right you will go to one of those Alex nets VGG nets one of them that has already been trained on millions of images.

Same thing right just as a word representation this word2vec it has been trained on millions of words coming from Wikipedia and all. Similarly you have something like image net which has millions of images it has been trained these networks have been trained. So, typically what you do is you know if it is an Alex net for example, you will go to the go to the last last, but one fully red FC layer fully connected layer right. I mean you have I do not know right. So, what is it like 4096 after that you have 1000 classes right.

So, that 4096 that layer right that layers representation you will you will actually pull out. So, what will you do you have the network with you and then all that you do is push your image inside you do not train just take that network it is all trained it has all the weights trained on image net. Now you push your image inside you will get some value that will be a feature vector right. So, that feature vector you will pass because that is far more robust than using intensities because that has encapsulated the correct information that should be embedded right about the right embedding for that image right. So, this x right.

So, when I say word to vector it does not mean that every time it should be a text it can also be an image in which case you will actually pick something like a feature vector you could also have you know a VGG network right within which again certain layers have been found to be good for style certain layers have been found to be you know right good for a content representation. You must have heard of style transfer and all right. So, there people use actually a VGG network again this is been it is not like VGG evolved like that it just that right people figured out that you know certain layer certain representation is better for certain tasks. Then H i let us say and then and then and there is and there is there is no sort of reason to have these dimensions to be same or anything H i can have you know a different dimension then O i again can have a different dimension R k let us say k and then what will be what will be then your then your U. So, U has to act on you has to act on x t.

Yeah. So, U will be of dimension. So, what is this D cross N D cross N right and then what will be your what will be this dimension of W. W. D cross D. D cross D right because it takes both I mean right what it outputs is actually H t and it acts on H t H t - 1 and so here right to give H t and therefore, both have the same dimensions like D cross D and what about V.

So, V should act on H t right and then give out O t. So, O t is. So, what is this what is the dimension of the output k cross D right yeah k cross D and then of course, you will have bias and all that right. So, again. So, that and all you are you know figure out.

So, this is what you have effectively then. So, the other one that that rolled up is actually a compact sort of representation this is called a compact representation of an RNN. If you show like this it means that right it means that you are that you are actually effectively implying this compact representation of an RNN. Now, there are right different flavours like I said let us let us just talk about a few of them. Let us say let us say I have you know image caption I talked about it already.

Now, in terms of in terms of the architectural diagram now say earlier we just wrote something as a output now suppose I want to go back and then talk about the architectural diagram right. So, what I have is an image that is coming in right and what I want is actually you know caption right that needs to come out a caption is actually actually a text right. So, the way right you show it is here is the image let us say I when I say image I mean I mean a visual feature of the image a visual feature a visual feature visual feature that means it could be you know whatever f c what f c 7 know f c 7 of an Alex net or whatever right v g g layer whatever it is it could be one of those representations not the image itself. And then then you actually then this goes as the goes as the right initial state in the sense that see for example, right I mean I know if you go back here I mean somewhere you have to start with some H naught right if you keep on going back. So, it is like it is like it is like this is like this is like first information right which you have and which is actually a summary of all that you have in fact right and this goes in and then out comes let us say the you know the you know first word right let me call this O 1.

So, it could be like right. So, it could be like you know the boy is playing right. So, O 1 will be like the right O 1 ideally should be like the and then right I mean and then right you can think about this summary being passed on to the I mean next guy and then right this will not have any further inputs because because the entire input is gone in here this is same image right I mean you can keep on showing, but then, but your state is already supposed to the hidden state is already supposed to encapsulate all that information there is no point again again you can afraid throwing it in. So, if that does it is job correctly then then then right you might have something like you know ok. So, let us say this O 0 is like H 0 then you have like H 1 and then out comes like O 1 and then right you may have another going this way and all this W and all are there ok U V W whatever and these dimensions are all right one has to match ok.

One is assuming that there is a matrix sitting there which will which will take care of the dimensional matching matches and all that ok and then and then you go on it O 2 and so on right. So, this is like one to many right. So, so like I said right there are actually different flavors it is a one to many one that is going in is just the image and what is coming out is like many many outputs which is like you know text running sentence or a caption for the image. Then you can have let us say another one is many to one ok this example is like a many to one example where let us say activity activity is sort of a classification or activity recognition whatever you want to call it classification. So, what is it like activity classification means what it could be like a like a video that goes in right somebody is doing something and then you want and then you have a bunch of classes as the output right it could be one of those classes right and therefore, the output is actually one class you have to say whether he is jogging whether he is sitting whether he is whatever right jumping frame.

So, what goes in is actually a bunch of videos as a bunch of frames. So, this is like a video that goes in and then right outcomes outcomes a classification score right where where the right action should get the highest sort of a probability and everything else ideally. So, this should be like a one hot vector right where only that particular action gets flagged and everything else is like given as 0 and here right. So, here we here. So, here what will happen is the first frame goes in right and then and then right it goes in to this vector, but then it you do not you do not you do not immediately output anything because you have to watch and that is why I said ok.

So, here you do not. So, so, so, so, so, it is not like the frame came in and therefore, right I have to now say something you need not because you have to wait then this then the summary information is passed to the next state and then and then again now this let me just write it as what are the first frame then the second frame then the third frame right all these go in and then may be right and then whatever n number of frames have gone in and then after you have accumulated all the past history right coming out of all the frames which is all which is all being passed through which is all being captured right through these states and finally, right here after you have seen all the frame like n is the last frame that you have seen from the video then after that right I mean you can have an output which should which should be this right which should be like one one class coming out of it to be like 0 1 0 0 something right some action that is happening. So, so, it is like many things going inside and then one thing right which which comes out. So, this is like a like a many to one then you can have you see many to many right many to many. So, they like a language translation right you have got like a sentence going in and then a sentence coming out. So, here how do you think with this this architecture might be like many to many how would you how would you want to get a build this architecture would you like to see the word and then immediately output something or would you like to see the entire sentence.

. Exactly right. So, you would want to watch the whole input because sometimes right the I mean it is not like when you translate we do not translate word by word right the language. So, if you are right in Hindi it is not like hello and maybe sometimes it is, but they are not always right. So, the structure is somewhat like this and this is called an asynchronous structure by the way. So, you have an input and then and then right and all this. So, so right each is a word by the way.

So, each is a word that is coming in from the language that each be translated right. Then you push all these things and now right you have kind of learnt a history of the entire of all the words right from the past and then then you start sort of red outputting. Then this goes here and then you know what sometimes right sometimes what can also happen is I think right that example maybe I think it is there somewhere else ok. So, something like this what will happen. So, this is like many to many and it is asynchronous because it is not like for every input right you have to actually output something and then the again the sizes can be different ok.

The input can have you know whatever a certain length certain number of words output can have a different you know length. So, all these are all these are words right as output and this could be a different language. So, this is like language translation. And again right I mean and you again need this because again the history of that language is again important. And I mean I write I mean if you are really wondering why do I have again this kind of a this kind of hidden state there right that is because the language into which you are translating that also has a construction right.

So, the history because there also if some certain word occurs then certain things will occur accordingly right. Therefore, you cannot ignore that right. So, that is the reason why this history is also coming in at the output because there also you want to capture what is what is happening and then you can also have something like this called an asynchronous kind of an architecture right. And then you can have you can also have a synchronous architecture right where asynchronous this is like asynchronous many to many you can have synchronous many to many where let us say where let us say right I mean you know you can have some like you know you know I give you a word and then you need to tell whether it is a noun or a verb or an adjective and so on right. So, for every word that I input the output you have to you have to immediately tell an output this is a proper noun is this right what is what is this kind of thing right.

So, you can have architectures like that. So, if you want I can just give you give you right one such example. So, which is again many to many, but then synchronous. So, right. So, that is why depending upon what problem you are dealing with the architecture can keep changing ok.

So, there is nothing like a fixed setting you have to figure out what fits best ok. So, this many to many it could be this synchronous. So, for example, right. So, you give you push this and then it goes through this and then out comes the output then again it this goes in and maybe and maybe the history again is important you know because that you cannot again ignore it. So, whether you might say that just by looking at that word I might be able to tell, but then maybe if you also looked at the previous word it might help you if you also looked at the previous history it might actually tell you you know in make your inference you can you can probably do a better inference right if you knew the history right.

So, again. So, it is like it is like you know. So, the output right could be could be a could be a word classification right could be a word classification as noun whatever right. So, where is it a noun right is it a verb whatever right something like that when you can have something like that and therefore, it at the output I mean every time right it will have to tell one of those classes which which particular thing it is right. So, that word is and again history does matter in all of this that is that is the temporal information is good to carry with you is good to encapsulate it and carry with you. Now, what about the loss function right what do you think what kind of loss function do you think really you will have here? I mean one also has to now worry about it how does one train this and so on.

We are now we are not going to look at the back prop and all that like I said. This is this use. One down to the. Yeah yeah yeah go ahead. As an asynchronous structure do we require a stop flag kind of.

Yeah actually you actually need a start and a stop flag typically there is something called go which is which is like that is when you start you start the start you know right throwing out the word and then stop is like you know when when you have ended the word. I mean I I did not talk about those details, but yes those are all there. This loss function right what do you think what do you see first of all what are the unknowns θ right what do you think are the unknowns first of all what do we have to learn u v w a b right this all we have these are the only unknowns that we have to train for. And our loss function right if you see the loss function is has to be accumulated over time now because each example now is like a say think about a language translation. So, what is each example that is going in each example is like a sentence right one sentence goes in another sentence comes out and there is there has to be some kind of calculation it as to how well have I done there and then then you then you could as read push in another sentence that has to be translated correctly push in another sentence that has to be translated correctly.

So, the loss right should be over over again of every example right for every example I should compute how well have I done and I should sum up all those all those losses right. So, this l θ right if you look at it. So, this l θ will have a form. So, till now we never saw this time and all right.

So, but that will come in now. So, t equal to let us say 1 to t l 1 of $\theta$where this l t of θ where this t actually means that at that time instant whatever example went in find out a loss for that and then and this loss can again be again the same thing. So, the loss can be a cross entropy loss cross entropy loss just as if you are doing a doing a classification problem it will be a cross entropy loss or if you are doing something like a regression then then it can be actually messy that and all will be exactly similar to what we had before ok. Cross entropy or this one and and the way right this done is what is called back propagation through time ok. BPTT it is called it is fairly involved ok I mean I teach it when I teach a deep learning course, but we will not kind of go into it here ok it is fairly involved ok. So, so what you do is just as in C-R-N it is also right I said that back propagation you can definitely do it is not so hard or you can probably look up there are so many so many people right that have actually written about it.

So, just go through those things sometimes of course, people can all right mess up the notations, but maybe there are some good places where you can go and see that how this is done.