Modern Computer Vision

Prof. A.N. Rajagopalan

Department of Electrical Engineering

IIT Madras

Lecture-27

Then the next one right that we want to look at is Google net, that is again I mean in terms of number of layers it is only significant I mean it is not very high I mean it is about it is a small jump from 90 to actually 22 and even in terms of the accuracy right it is like it is like a small improvement at 7.3 to 6.7 but this improvement is also you know we cannot ignore it because as I said as you go downwards right it is going to be harder and harder. But there are certain things right that a Google net brought in you know which have been then used across the board certain ideas that they brought in which of course helped the image sort of you know classification challenge but there is something else right which also you know it triggered in terms of you know understanding what is called what is called an SE Inception module. Okay and this Inception module is now is now kind of widely in use. What their argument right everything you know so their argument hinged on actually you know something called a 1 cross 1 this one a convolution.

I mean you know see for example if in you know DSP course or in a signal and systems course we told you that I have a 1 cross 1 filter right what would you think right it can do if I kind of pass an input and I have an impulse response which is 1 cross 1 what will happen to the input? It will just scale it right I mean all that can happen is either if you just have if you do not have any I just have 1 there then the input will just come at the output if you have some number there a times it will just become a times right can a be a complex number? Yeah right it can be a complex number also right so whatever it is right whatever a times delta t and then right xt goes in yt is simply a times xt that is what will happen. But this 1 cross 1 convolution is not that okay it is also it is also like you know one a single sort of a pixel alone but then thing is right it is acting along the along the SE feature maps. So what is called a depth slice okay that is another another sort of you know terminology to use I mean I keep using feature maps but some people will say depth slices okay so you have you know the number of feature maps if you say that I have a depth slice which is higher that means I have got you know deeper I have got more number of feature maps okay. So this 1 cross 1 this one a convolution acts along the depth and the idea is that at any time I mean not just for this example okay but the idea is very significant because this 1 cross 1 this one a convolution you can use any time you want to actually bring down something like a like you know a big volume that you have with you if you want to convert that into a 2D feature map right you can always do it I mean you can just use this 1 cross

1 this one a convolution and then aggregate all the information right bring it down to you know a 2D feature map.

If you want a 3D feature map to come out that is also possible you just have to use multiple multiple box filters but the idea was that you know you could the first thing is that using a convolution a 1 cross 1 convolution you know which I think you know till then probably nobody even thought about a 1 cross 1 convolution because generally we do not talk about it but here the idea is that the filter will have different different values for each feature map right as it acts along the you know depth and so right that way you still have a filter a box filter which you can then translate all over what will happen when I have a whole volume there and then you know I am kind of right sending this box filter all around it will give me one 2D feature map. So they said that right this way you can aggregate all the information bring down the number of unknowns that you have or for example otherwise you know you would have to not just the unknowns what I am saying is the number of feature maps right which you have to actually deal with you can aggregate all that information. Now whether you lose out right in the process whether you gain and all right I mean you know it depends upon where you actually put it and then how you use it and so on but the way but the way right in this thing okay in this you know Google net the way right it was run by the way this was by Zegidi 2014 so as you can see right these are not all that old and all I mean by our time scales right these are all pretty new that way but then so much has happened I mean the funny the sort of strange thing is that you know even though 2014 looks like just about 7 years 8 years ago but so much has happened right in the last especially in this area right every day tons and tons of these papers get uploaded and archive and everybody is trying out something therefore the rate at which things change is very very fast okay. So 2014 is now old by the way okay so this is the key idea was this right efficient right inception module and if you see so this so the inception module had actually right you know two things I mean I am not writing these things but maybe some point right I will still write just for the you know sake of so if you write if you think about Google net okay some of the some of the main things right I will just write down okay why so this so the key ideas one of the key ideas was this inception module I will write and then I will actually explain okay about this should do want me to write bigger or this okay. So right use so the idea is what use multiple filter sizes in parallel see multiple multiple filter sizes right we have already seen like for example you could have had 3 cross 11 cross 11 then we use something else right 5 cross 5 somewhere along the line but then in parallel right whatever we did till now was all like sequential now one layer will do will do a 3 you know 11 cross 11 convolution then the layer following that may do a 5 cross 5 then the layer following that may do a 3 cross 3 but then in parallel right we did not see till now like for example you know you have all these filters coming together it is in parallel is very important multiple filter sizes okay multiple filter sizes coming together in parallel and then there was a concatenation idea right which was again

sort of new.

See concatenation in terms of you know combining you know that is multimodal inputs was kind of known but then concatenation of feature maps and all right which was still sort of you know a novel idea so this right so this is like you know act together okay so this was they all had to act right simultaneously so multiple filter sizes so what they introduced was 1 cross 1 3 cross 3 5 cross 5 and then in fact they even had had a you know a max pool block so they had all these all these guys together I mean but normally right you would have seen them acting sequentially max pool coming after some time after a convolution block and so on but in an insertion module we see that they are all kind of coming together in parallel okay and this 1 cross 1 is also called a bottleneck layer okay bottleneck bottleneck layer because you know because it actually reduces the dimension of the feature map I mean not spatially but then depth wise so it is like it is like it is like having a having a highway right that is that is that is that is that is wide and then you suddenly shrink it to a single feature map right that is like that is a bottleneck you know I mean otherwise you would have had so many of these channels right and then you are suddenly right you know bringing them all to constricting them to a narrow sort of a lane right so reduce the dimension of the feature map so this bottleneck layer is often used when you talk about autoencoders and so on but even this 1 cross 1 right people you know it is not unusual to refer to them as you know bottleneck layers okay now let us just go back to that figure right and see okay what what it is so here we were okay so if you see right what what they have is a previous layer output okay which is which is which is whatever was was earlier of course you know and you also see something here I mean so the inception modules are all sitting here and as you can see this is one inception they have got several of them okay so they have got several inception modules and then and then right now right let us not worry about worry about this and this let us just kind of look at the inception module okay so if you so if you look at the inception module right it is kind of see taking the output from from a previous layer that means right so for example this guy gets input from a previous layer and that layer okay and and then what happens is you actually actually reduce a 1 cross 1 this one a convolution but then what then what you do is you know it is not it is not like right you completely reduce it to actually a 2D layer so what they do is they use multiple box filters again 1 cross 1 convolution but then they they reduce a number of feature maps typically it will be like half the number okay that you might have had here so for example if you had 512 feature maps then very likely that you have got to see right or 256 feature maps here that means you have got like you know 256 box filters 1 box filter will give you 1 feature map you know 2D feature map you have like 256 of them I am just giving as an example so you have you have much much fewer numbers here as compared to what you have here in terms of the number of feature maps and similarly right you have something here 1 cross 1 convolution and then and then this and then right you have some kind of you know you know a max pooling okay so max

pooling is also is also a part of the part of the right inception module and then after this right what they do is then you have you have a convolution filters of kind of various types so you have a 1 cross 1 convolution which is coming straight away from here so so so this layer in the inception module it has actually two things right a bottom layer and then you know top layer in the bottom layer right I mean you have two 1 cross 1 convolution coming from here and again these filters are not identical okay the box filters that are doing the job here are not the are not identical to the box filters that are doing something here okay so these are to be learned separately max pooling no unknowns to be learned there and the second layer is all simply a convolution but then you know like I said multiple filter sizes so got like 1 cross 1 got 3 cross 3 5 cross 5 and then 1 cross 1 okay and again again right I mean the say right number of number of feature maps and all they have I mean if you kind of if you if you see the architecture it it will tell you how many feature maps is there out here but then you know in terms of the overall overall numbers okay if you if you were to if you were to do a do a naive job see for example right I mean if I try to use a previous layer from here and then and then and then right if I tried to if I if I if I skip this guy and suppose I directly operated here right then then then the then the number of feature maps right that this would have to operate on will be much higher because whatever is here right this 3 cross 3 convolution has to act on all of them so so this in between layer is actually bringing down the bringing down the depth okay significantly so that so that these filters when they act then they act on a smaller depth okay and then this sort of a concatenation idea of course to concatenate one of the requirements is that they should all be of the same size I mean you can't concatenate different sizes okay so concatenation means that you just put them stack them all together so what you will do is whatever output comes from here so it will be a set of feature maps then you actually again you have to zero pad and all properly that's where you know by till now we didn't use zero pad that effectively right it just look like okay if you wanted to do something this uses zero padding but here is where if you wanted to concatenate feature maps you can use actually zero padding is one one one sort of tool that is available to you wherein you can actually adjust the padding such that what comes out of you know one box that same dimension you can actually maintain across the block so so you would have to use appropriately again here again here and then right and then okay this is a one one cross one convolution and you will get again feature maps corresponding to each coming out of each of those blocks and then you stack them all together okay that is a concatenation and that concatenated output is what then goes as goes as input to the same right next layer and so on so the key thing okay two kind of key things one is multiple multiple filters acting in parallel multiple filter sizes acting in parallel which you didn't see till now second idea is this one cross one this one a convolution which again was something that that you know right we did not see before and it turns out that having an architecture like this even though this is not very obvious for example if you and I were to sit down and think about okay what might be a good improvement it's not very obvious that we would have struck upon this idea see it's also

something you know right it's not like very obvious that you know that you know right one would have thought about an architecture like this but then the fact is you know their group worked on it and then they came up with this architecture and this one right this called this called an auxiliary okay there is there is one thing okay which which which which you know which of course you know I couldn't I couldn't talk about for lack of time I think there is something that I just mentioned along the way there is a vanishing gradient problem okay in in a deep network okay there is something called a called a vanishing gradient problem there is also there is also an you know exploding a gradient problem but typically you know a vanishing gradient is what is most common so what this means is that right by the time you actually propagate the you know gradient information back to the input right because you know right from all the way from the loss at the output I mean you have to come all the way to the input now you are hoping that your gradient will still be significant right that is when the weight update will happen right if you go back to that you know GD okay you know that it's like you know theta n plus 1 is equal to theta n minus some alpha times dou L by by right I mean dou this one theta evaluated at theta equal to theta n now that you know that there is a gradient dou L by C dou theta which you can of course write in terms of all the or involving the right I mean all the all the slopes right in between so when you come all the way from the output to the input right this this number okay you can show that sometimes if you go deeper and deeper it this gradient can become very very small therefore even though right you are having a gradient which is not 0 but then the gradient becomes so small that it almost vanishes therefore when you try to apply an update right the update either can become very slow or or it's it's almost it's almost not there it's going to say negligible which means that you are stuck right I mean you can't even move and therefore right people so in this network right what they have done is they have added an an auxiliary network which has some very very simple blocks but this auxiliary network right has and so right in between I mean it's not everywhere if you notice it's not in every inception module it's been thrown in right I mean here and there right so it's thrown in here it's thrown in here and the idea is that right so the so the idea is that idea is that you know this the final classification loss right is also is also recomputed okay at these places okay I mean okay we don't want to spend this right you know too much time okay on on on Hawaii and how it works but then the general idea is to handle the vanishing gradient problem you also you also have an alternative way by which by which you know the the by which the gradients do not do not simply right you know go down to zero so this auxiliary auxiliary network is mainly for that it just wants to wants to make sure that because because you have so many layers at how many this is about C 22 right so until now among the among the deep networks that we have seen 22 is still the highest the next one that we are going to see is far higher that's like 152 layers or something and there of course this vanishing gradient problem will will look like an obvious thing that will happen but even with you know right 22 layers okay it could still be a problem by and therefore this auxiliary network right I'll just write write a few

sentences about it and so that right I mean for those of you who are more interested you can go and read it up so this auxiliary network auxiliary network this is all for Google net okay by the way so auxiliary network okay so to so this is mainly needed to improve the strength of the gradient needed to improve the strength of the gradient strength of gradients which means which means that right now it has a classification loss by the way okay that is why it is able to improve the strength of the gradients then okay the concatenation is okay so zero padding I think and I've already said huh then the other one is a stem network I mean that's like a preprocessor okay see this one looks like a stem right I mean in the bottom okay what you saw here it looks like a stem right so this one right this is this looks like a stem okay so it's called literally a stem you know network and that's that's also I mean that's nothing very significant let me say that if I have if I if I can blow up and show you what it has I think it just has you know a corn layer a batch max pool and then a batch norm or something like that very simple stuff okay so the stem network that is okay this is a pre-processing module before the first inception module okay this is like a preprocessor preprocessor before the before the inception module before the first inception module okay not not not everywhere okay before the first inception module and yeah we can see I mean right what it has and there is one more thing right which these people introduced and we have seen max pool and all right now what they also introduced was so what happens is okay towards the towards the final stages right you do what is called global average pooling or what is called gap see it's hard to hard to right keep in mind all the of the feature map sizes and all but just what matters is this idea global average pooling so what what this what this actually right does is I mean towards the end that you can actually check this okay with respect to google net so when you when you are when you are done with the last convolution layer at the last convolution layer okay that's not coming out in that figure okay when I when I blow it up I don't think you can see that the last convolution layer you have feature maps which are of size 7 cross 7 cross say 1024 okay that's the size you have that means you've got 7 cross 7 spatial dimensions and you've got 1024 such feature maps okay now if you try to if you try to unroll this right and try to then put in a put in a fully connected layer after that right that will simply blow up correct because you've got like you know here itself we got 7 cross 7 cross 1024 so what they do is unlike the max pooling and all which we saw there is also something called average pooling right that I mentioned and we can have 3 cross 3 average pooling and so on so instead of that they do what is called a gap okay this is called gap okay gap so this global average pooling what it does is it kind of it takes the complete average right so here you have a 7 cross 7 filter it just which is boils it down to one value just takes the complete average of that feature map right so it's called a global average right no longer like local average pooling or max pooling that is locally happening right so they just just take the average I mean you can you can ask could they have done a max pool on the entire 7 cross 7 maybe right but then what they advocated was this okay again right and all of this there are so many things in which one can do okay just that we are only trying to tell what they have done okay so

what they did was so they just right reduce it to one number okay reduce it to one one value through gap okay and and that way right that way that way this whole thing could have say it reduces to reduces to you know 1024 dimension instead of 49, 49,000 now it is just right 1024 and this 1024 then kind of right goes into the you know fc layer that is why if you look at look at the see parameters right this this guy has far fewer okay so if you look at the look at the number of parameters let me see right where I've got the parameters for google net I think it's about 5 million or something if I'm not mistaken let's see in the figure somewhere it should be there yeah only only 5 right million so you can see right so what has happened is right towards the final thing once they did this global average pooling right so just this just shrunk down to down to 1k and after that right they had a few you know fc layer fully connected layers but all of that put together still is far far lower than you know so it is actually 12x less than alex net so so you should you should see see see the power there right I mean there you had 60 million and you're getting 3 point what is that 6.7 percent of error with just 1 12th of what you had used for alex net right so that's also the reason why this network is actually a powerful event because again you know it's hard to kind of pin it down and say you know which one is you know which one probably was you know how to account for each one of those how to account for this for this improvement right through each one of these modules is hard to say but you can definitely but overall right the sense is that getting this insertion module is is actually a key and then this global average sort of a pooling where maybe right people were afraid to do right prior to that they thought if you do it something like that then maybe you lose out information for example at such averages and all when you do a regression problem typically you don't you don't do such kind of things because because what you need a finer details and any averaging rate will sort will will will make you lose information so not usually right so you won't kind of think along those lines but these people went ahead and you know so at the end they kind of did that and then they showed that with just 5 years here right million this one parameters right they could do so well okay then is there anything so inception model i think i already told you one cross one convolution i told you concatenation i told you right i think all this is there in the slide i think you know it's easy for you to understand okay and i think the so so for example so if that if that in between layer was not there right the inception model had one thing here right one cross one one cross another max pool without that right and there is also this what if you had a naive inception model without that then what would happen so i think right these are all things okay which which are now easy to understand for you and you can actually find out okay and what you gain and what you lose all that is here okay now let me let me go to okay so this is a naive model which if you use right you will actually run into trouble so this is the one that you that they use you know which leads to a dimensionality reduction then bottleneck layer i think i told you yeah i think you know i i thought it's easy to explain first and then the stem network right i told you what else is there auxiliary classification to actually inject gradient at lower layers i think most of it is here.