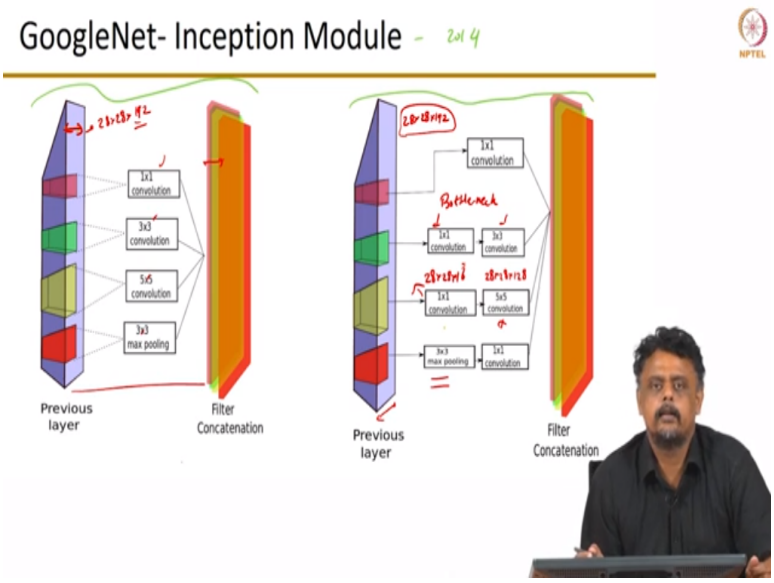
Machine Learning for Engineering and Science Application
Professor Dr. Ganapathy Krishnamurthi
Department of Engineering Design
Indian institute of Technology Madras
CNN Architecture (Google Net)
Part3

(Refer Slide Time: 00:12)



In this video we will look at Google net, the basic building block of this Google net is the inception module again the 2014 winner of the imagine a challenge, so let us look at the basic building block of the inception model so for instances he had for the VGG network that we saw in the previous video the basic building block was a block of convolutional layers so see very small filter sizes and if you looked at wave (())(00:47) Alex net, Now Alex that had variable filter sizes, so each first layer and eleven cross eleven and then we had five cross five filter and the we had three cross three and then a fully conditioned, so the inception module incorporates both this concept since that every layers has all possible filter sizes, so they build a clock a convolution block which had multiple filter size and we let the network the backdrop the learning to which limitless and let the backdrop decide which weights to update based on your objective function.

So let us just quickly look at this inception module so if you look there are two images so this here is what the authors of the Google net paper called the Naive implementation and this is the actual implementation that they are followed so what is the different so we have an input feature

map this is thickness of the feature map right here that means a different color, so this is the thickness of the feature map as I have pointed out of here the size some K feature map and we have one cross one, three cross three, five cross five and three cross three max pooling also applied to this feature maps the outputs each of them are taken and concatenated so they are gone getting it across.

So again this means that we have to do that convolutions so as to get the same sized filled feature map from each of this convolutions so there are all zero padded accordingly and to get the correct size which are enough, so that they can all be concatenated, as you will see doing this as a problem because the number of computations becomes huge so if you just show this is the naive implementation so but doing it this way, Implementing the block the inception module as they call it this way has issue in term of the number of computation involved it becomes huge because we have large filter sizes five by five also in that, we can also use I think it is possible to use seven cross seven also but the authors of the Google net inception module just stop with five by five probably work very well for the image challenge okay.

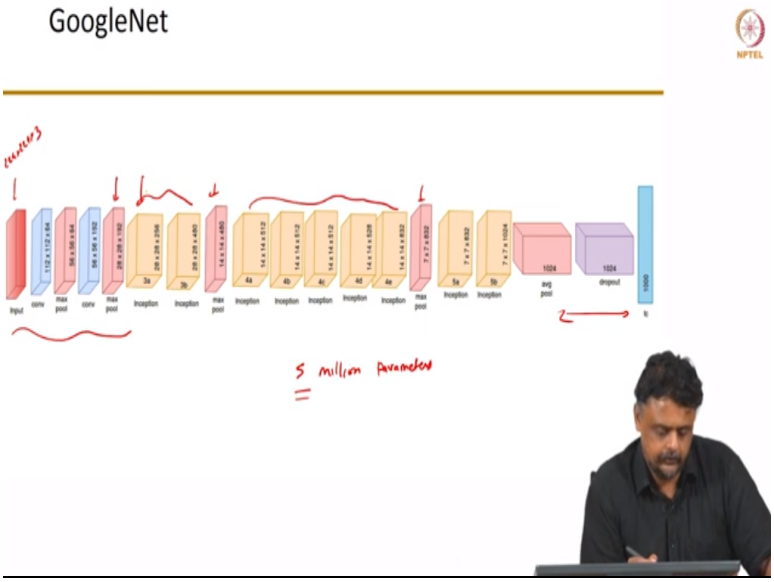
So what does the better implementation, the better implementation is to use one cross one convolution to reduces the size of the filter maps, since when I say size in this context it is the depth of the feature map, so if you are input layers has let us say, the input layer has 28 cross 28 cross let us say 192 inputs so many feature maps so in this case I mean I had it down here so that is where we are going to do the one cross one 192, so we would use the one cross one convolutions so project this number, number of feature map to a smaller number to a smaller volume so what I mean by that, we know that as if you recall we know that one cross one feature maps to preserve the size of the feature drops in essence in plain the XY size of the feature maps are preserved but we can use one cross one convolution to reduce the depth of the feature map so that is particularly precisely what the inception model accomplishes so, so if you have a very large in a sense of a very large number of featured maps in the input volume you would use this one cross one conclusion to bring it down so for instance in this case.

Let us say we take the one class one convolution before the 5 cross 5 we can use the one cross one convolution to reduce the size of each other up to 28 cross 28 let us say 16,and then we can output let say from the phi cross phi convolution so you can output 28 cross 28 cross 128 feature so just to recall the how this is done is by defining 16, one cross one convolutions of course one

cross one grand illusion act across the volume across the featured maps so you can also question whether this would you know lead to loss of information so this seems to be a parameter they have tweaked the number of feature maps here this has to be done by some kind of cross validation to see which is the most optimal so this one cross one convolutions prior to doing the larger convolution with in a sense larger convolution in a sense of conversions with larger filter sizes so the one trust one project the input volume to a smaller dimension and then subsequently you use three cross three convolutions to put them back.

So this is what is referred to now as bottleneck, bottleneck layers so you string the size of the feature map that way it works similarly for the max pooling also you can do because max pooling would in most preserve the size if the feature maps the depth of the feature maps that is and you can use again one cross one coefficient to reduce the size so this was the begin the two of the big innovation in an inception module so instead of just so if you saw in VGG that each block they decide a divide the network into block so each block had a succession of condition so in this case for inception module it incorporates multiple conclusion kernel sizes just like in AlexNet we saw that we used eleven cross eleven and five cross five, three cross three but in this case it is five cross five, three cross three, one cross one and a maximum layers all in one wall in one module and you also use this bottleneck one cross one convolutions to project the input volume to a lower dimension in terms of the depth of the volume before we do the three cross three column and the other conditions with higher filter kernel size.

(Refer Slide Time: 07:23)

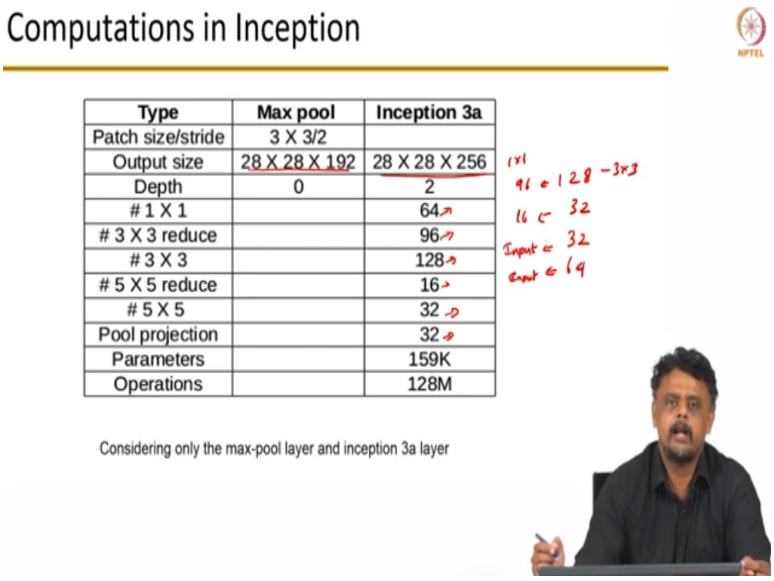


So you just look at the network this had about twenty two layers with mates and so there was an initial set of convolutions and max pooling which reduced the size of the input to 28 cross 192 so the input as usual was a 224 cross three it was followed by sequence of convolution and max pooling sequence of conversion and max pooling to get it to 128 was 192 and then that was used as input to the inception layer sequence of inception layers followed by max pooling again sequence of inception layers followed by max pooling so on till we have the typical output with one of thousand activations, so it had so they have labeled the inception module as three for B this was the if you read the paper I urge you to go read the paper where see that they have labeled each of these convolutions the inception modules by you know 3A 3B in this case for 4A 4B and there is corresponding cable which tells you how the computations are done in that particular module.

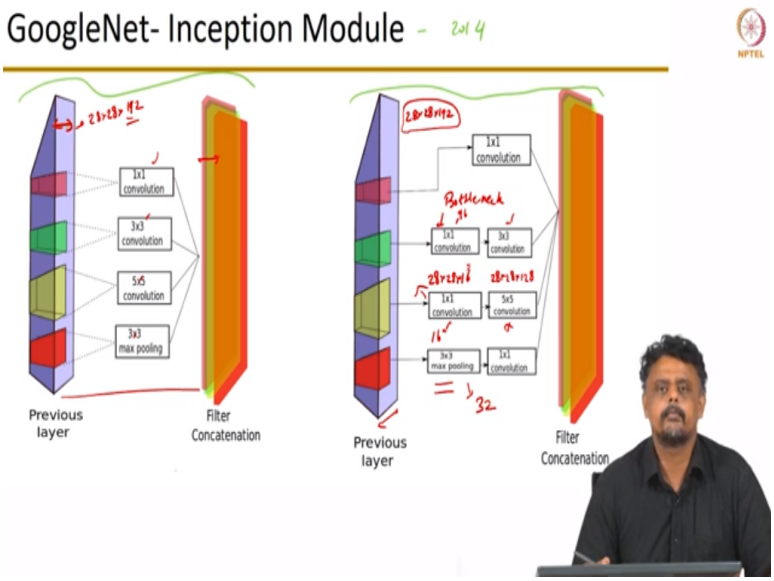
So we will just walk through one inception module that is 3A and see how the number of the saving in the number of computations by using a one cross one bottleneck to reduce the size of the field reduce the number of feature maps so this is a 22 layer network but then it had very few parameters about five million parameters and it with 1D 2014 meeting challenge which I thought for error rate of about 7 percent it is slightly better than VGG but with much-much lesser number of parameters so this was considered this again is one of the networks which have and if it is very small not very deep but in terms of number of parameters very less number of parameters

the contrast to let us say Alex net so that we need G media are originally 1638 million Alex not at 60 million parameters but this only had 5 million parameters south weight network so we will look at this inception 3A right here which takes us input 28 cross Ronda cross 192 and the output is think 256.

(Refer Slide Time: 10:04)



Type	Max pool	Inception 3a
Patch size/stride	3 X 3/2	
Output size	28 X 28 X 192	28 X 28 X 256
Depth	0	2
# 1 X 1		64
# 3 X 3 reduce		96
# 3 X 3		128
# 5 X 5 reduce		16
# 5 X 5		32
Pool projection		32
Parameters		159K
Operations		128M



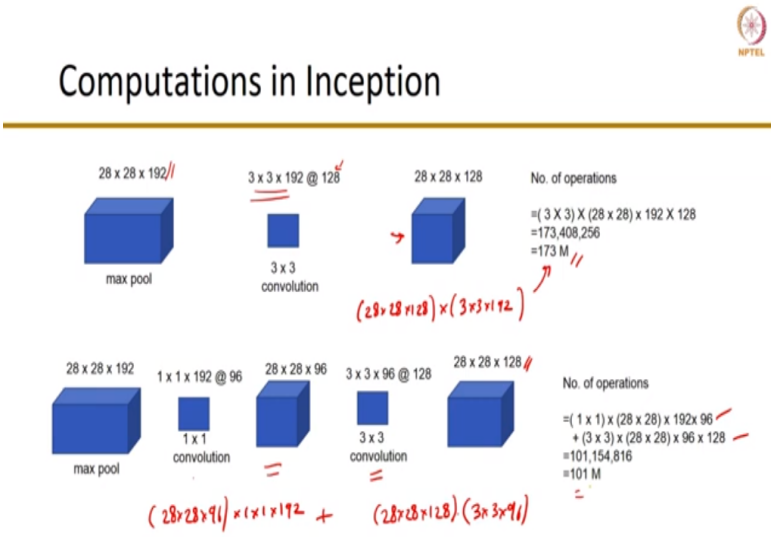
So just look at it so we reproduced a piece of the table here I urge you to go back and look at the table so this is the output size of the max pooling layer which feeds the as input to the inception is the input tween to the inception module three and this is the output of the inception model

three.so it has 64 one cross one convolution so we saw that if you go back and write it down there just to see and then it had 96 three by three reduce in that table means the number if one cross one convolution done the 96 feature maps were produced by the one cross one convolutions prior to doing the three cross stage so you would produce 128 feature maps with three by three again 16 one cross one feature maps produced by the one cross 110 Aleutians following by 32 and then 32 from the max pooling layer, so the output of the max pooling layer if you go back if you recall some of these numbers so we would do here so what it mean here is the reduce is basically here so prior to three cross three we would have 96 feature maps here and price to phi cross five will have 16 feature maps and there then the output of the one cross the max pooling would give you give you we sought 32 feature maps,

So and if you look at the output of the three cross three conditions I think this has about if can go back and look I do not want to go back and forth again so the three by three produced about 128 and if I cross five produces 32 right the pooled projection layer produces 132 there is a max pooling layer and the one cross one convolution it sells produces 64 this is from three cross three of course there is 96 coming in to this I am not mistaken yes the reduce is 96 and reduce is 16 here but this is the one cross one convolution this is the three cross three if I classify and for the pool projection there is nothing for the other one there is one cross one convolution this is directly from the input these two are from the mean.

So the hash three by three cross three is reduce the number of feature maps produced by the one cross one condition recall that I will once again back here recall that the one cross one condition are done prior to the three cross three and the five cross five and the one plus one convolution following the max pooling there is no nothing before then there is one plain one plus one convolution layer, so which is what we have here so this is 64 one cross one 64 feature map stories from one cross 196 by the one trust one prior to the three cross three and then output from the thick roster is 128 and output from the five cross five is 32 but prior to that you reduce the dimension to 16.

(Refer Slide Time: 13:34)

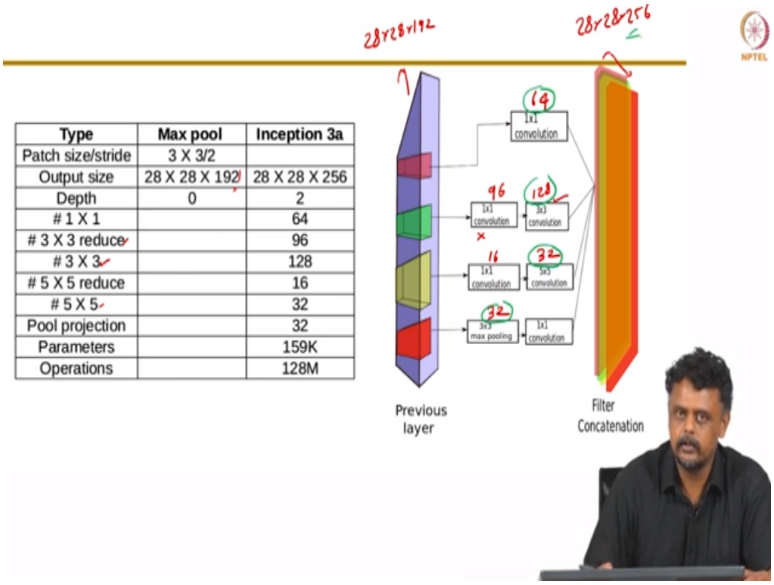


So what is the savings in terms of nominal computations so if you look at the following output of the max pool which is the input to the inception module is 28 cross 192 now if you directly do 192 three cross three convolution of course then the size of the number of parameters in every filter is three cross, they cross 192 and then we produce let us say 128 feature maps that is 28 trust one day trust 128 then the number of operations would be about 173 million, now if we do let us do the one cross one projections into a smaller volume if you do that then we produce 96 feature maps following the one cross one condition and then we follow it but with the three cross three convolution to again produce 128.

So we have to do this calculation here for the one cross one convolution so the easiest way to this is again very easy to write it down I will write is down for the thing and then the ok, so if you have to do three cross three convolutions to produce 128 feature maps of size 28 by 28 so the number of elements in the output feature map or the number of activations note which out map is so much right and then for each feature for each output activation we have to do how many computations three cross three cross 192 of products right so three cross three cross 192 that is pretty much what you see there ok that comes to one by seven remaining you can repeat the calculation here, here for this one too so the output feature map here for the one cross one contribution is 28 times 28 times 96 this is the total number of activations produced for each activations we have to perform one cross one cross 192 multiplication so that is what we have here and then similarly for the three graphs take on illusions we produce 28 cross 28 cross this

plus 28 cross 128 right and for each one of these activation we have to perform three cross three cross 96 multiplication, so that is number is here we add you see that there is a reduction in the number of computation that you have just for this one particular feature map set of feature plums, so this was the innovation behind the inception, so it does two things it let us the network decide which feature maps are more relevant based on the back propagation or the optimization at same time she saw earlier again even with VGG net we saw that using larger size of type field means that the number of parameters increase in number of computations also increased correspondingly, so that is solved by using bottleneck layer using one cross one convolution where one cross one is used to project your feature maps to a smaller dimension here the reduction is along the depth of the feature map so the volume becomes smaller.

(Refer Slide Time: 16:55)



Type	Max pool	Inception 3a
Patch size/stride	3 X 3/2	
Output size	28 X 28 X 192	28 X 28 X 256
Depth	0	2
# 1 X 1		64
# 3 X 3 reduce		96
# 3 X 3		128
# 5 X 5 reduce		16
# 5 X 5		32
Pool projection		32
Parameters		159K
Operations		128M

So we will look at the inception three a layer, so the table is reproduced from the paper it tells you the number of one cross one three cross three convolution etcetera, in every layer, so just let us go one layer by layer the input to the inception three a layer's of size 28 cross 28 cross 192 layer the output has same size 28 cross 28 cross 256 so now it has number of one cross one convolution is 64 so there are 64 here, The hash three by three reduce refers to the number of one cross one convolution maps produced by the one cross one conversions preceding the tigress three to which means that this one cross one convolution layer will produce 64 feature map of the same size G for feature map and the three cross three convolutions will output three cross three ground water put 128 so that is are the two the three hash three cross three reduce here refer

to the convolution layer one cross one convolution layer preceding the three cross three convolution.

So it refers to this one right here so the number there the number of feature maps produces as output there is 96, these serve as input to the three cross three convolution to the three cross three produced an output of 128, which is concatenated there similarly the hash five cross five reduce refers to the number of feature map from the one cross convolution at 16 and the five cross five itself produces 32 feature maps the pooled projection layer produces 32 followed by one cross one convolution so that remains fixed in this case so the total number is if you add these up if you add 32 these numbers these are the so if you put these together you get about 256 output feature maps so this is just for one inception block so you can work through the table in the paper and see if the current calculation are consistent with the structure that I showed you earlier.