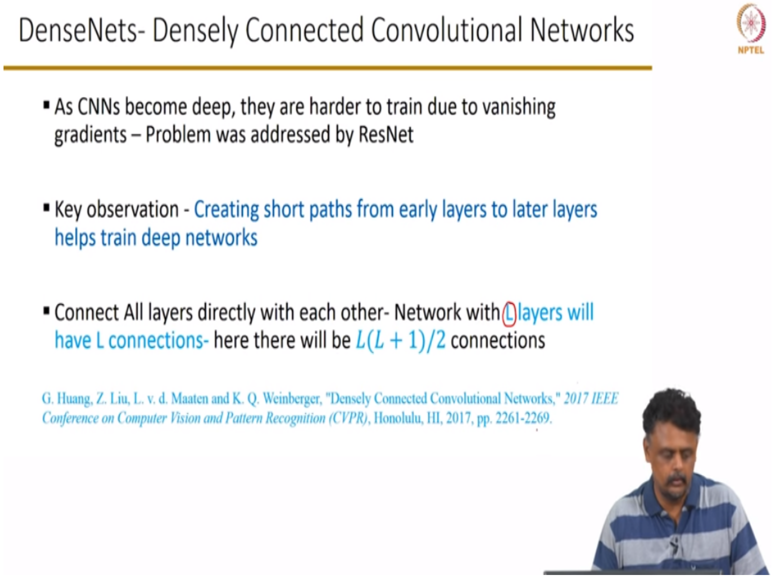
Machine Learning for Engineering and Science Application
Professor Dr. Ganapathy Krishnamurthi
Department of Engineering Design
Indian institute of Technology Madras
Dense Nets – Densly Connected Convolutional
Part 5

(Refer Slide Time: 00:15)



DenseNets- Densely Connected Convolutional Networks

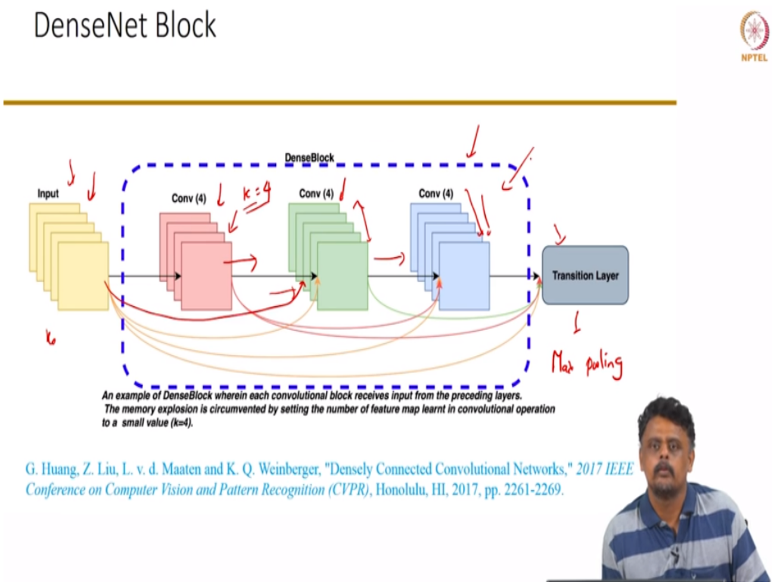
- As CNNs become deep, they are harder to train due to vanishing gradients – Problem was addressed by ResNet
- Key observation - Creating short paths from early layers to later layers helps train deep networks
- Connect All layers directly with each other- Network with L layers will have L connections- here there will be $L(L+1)/2$ connections

G. Huang, Z. Liu, L. v. d. Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017, pp. 2261-2269.

In this video we will look at dense nets one of more recent architectures was used in the image net classification challenge and had shown exceptional performance in terms of classification accuracy despite having a fewer number of parameters just as we with rest nets as the scene in the game deep becomes harder to train because the gradients begin to vanish this public particular problem was addressed by rest net by adding feature maps from the previous layers when skipping a layer and adding them to the next layer but in general the key observation that this paper make is that by creating short paths from the early layers that is your layers closer to the input to the later layers these are basically layers closer to the output the gradient propagation is improved and so is the classification and in fact we contain very deep network more than hundreds layers by adopting this particular trick.

Now what this instance do is architecture it improves gradient propagation by connecting all layers directly with each other so suppose we have capitol hill number of layers, so typical network with a layers there will be L connections that is connection between the layers however in a dense net there will be about L into L plus one by two connections we will see what we mean by that in the next slide.

(Refer Slide Time: 01:41)



DenseNet Block

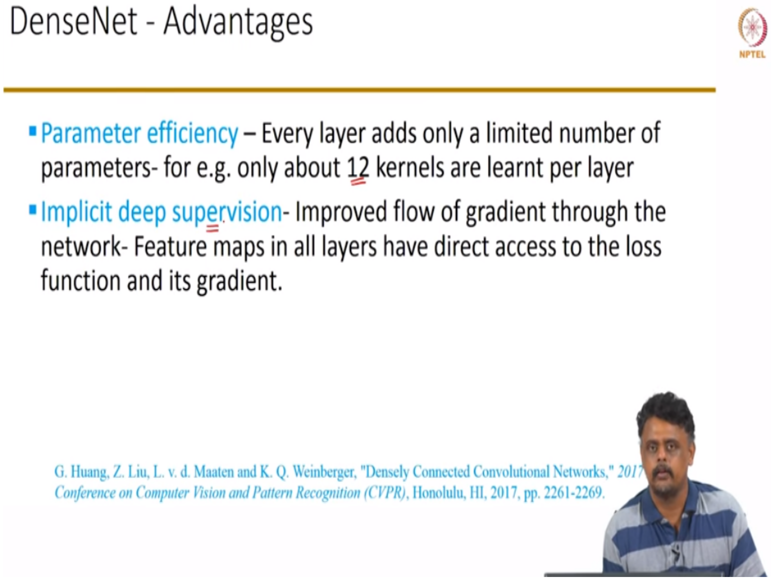
An example of DenseBlock wherein each convolutional block receives input from the preceding layers. The memory explosion is circumvented by setting the number of feature map learnt in convolutional operation to a small value (k=4).

G. Huang, Z. Liu, L. v. d. Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017, pp. 2261-2269.

Here is a particular incarnation of your dense net, so the inputs here there will be K0 let us say input maps again for an RGB image that like the one used in let us say the imagine a challenge that will be about three channels now the first layer creates a feature maps in this case K is for it is a feature maps but as you can notice as we go deeper into the network if we go to the second set of layers he takes as input not only from the previous layer but also the input layer so that is right there and then as we go to the next layer the particular layer here takes as input both the place all the preceding layers feature map of the preceding layers however the output of each of the layer is fixed so in this case there are about four feature maps of course we see that there are about five each map in the succeeding layers that is typically fixed and other thing that we notice is that as we go deeper into the network this becomes kind of an unsustainable, so let us say we have about ten layers then the tenth layer will take as input all the feature maps from the preceding nine layers, now that if each of these layers let us say produce 128 or 2 for this which are maps and these is a feature map explosion,

So do not work on this problem of course what we saw they fix the number of output maps from each of the layers and also created these so called dense blocks as we see here when this red or blue outline so each dense block contain a pre specified number of layers inside them and in the eye and among those layers the feature maps are shared like we discussed before and the output from particular dense block is given to what is called a transition which uses like a bottleneck concept like we saw with resonate and inception we uses a one by one convolution followed by a max pooling to reduce the size of the feature maps so this serves two purpose because now we can do max pooling so the transition layer allows formats pooling which typically leads to a reduction in the size of the feature maps now if you did not have this dense block kind of structure then max pooling would not have been possible because the size of the feature maps across max pooling would be less and it will be difficult to concatenate the feature maps if you just across layers.

(Refer Slide Time: 04:07)



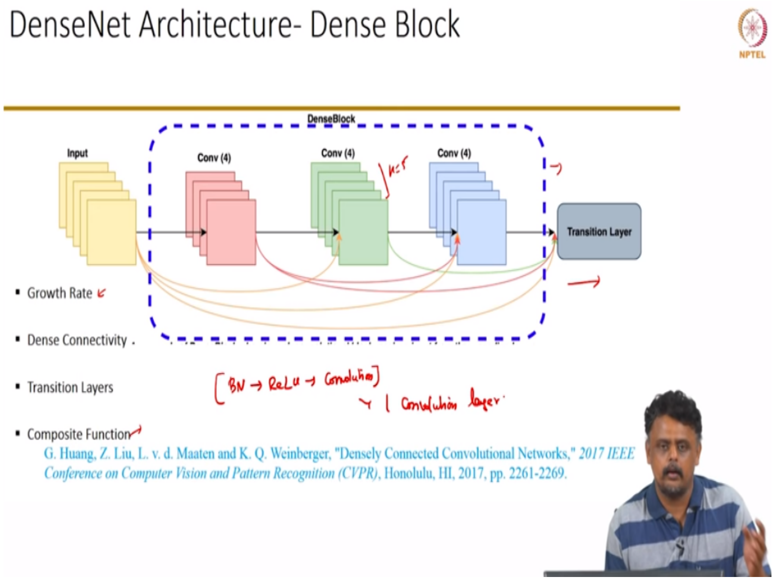
DenseNet - Advantages

- Parameter efficiency – Every layer adds only a limited number of parameters- for e.g. only about 12 kernels are learnt per layer
- Implicit deep supervision- Improved flow of gradient through the network- Feature maps in all layers have direct access to the loss function and its gradient.

G. Huang, Z. Liu, L. v. d. Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks," *2017 Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017, pp. 2261-2269.

The following advantages are proposed by the authors as far as dense nets are concerned use parameter efficiency, so because you have fixed number of output feature maps per layer only very few kernels are learned per line for example about 12 kernels this is one of the architectures they have suggested and then other architectures they are suggested 24 or 32 kernels per layer they also talked about implicit deep supervision and feature reuse, so what is implicit feature view super-efficient deep supervision, so for instance we saw an inception that they used auxiliary cost function using feature maps from the intermediate layers what that does is improves the learning in the sense that it has the if we just learnt have to be discriminative so as to improve the auxiliary cost function there have been several other approaches like that for us and there is one approach wherein you take feature maps from the intermediate layers, and give it to an SVM as input and it does the classification task and then that error is back propagated however here in this case as the feature maps are concatenated from the preceding layers the feature maps from the react fusion ramps are the activations from the earlier layers have a direct access to the error function or the cost function of course because we saw that these layers are grouped into dense blocks as they call them, so they are separated from the error function by a couple of dense blocks but they still have the feature maps or the activations have direct access to the error function there by improving training as well as learning discriminative features.

(Refer Slide Time: 05:41)



DenseNet Architecture- Dense Block

- Growth Rate
- Dense Connectivity .
- Transition Layers
- Composite Function

[BN → ReLu → Convolution]

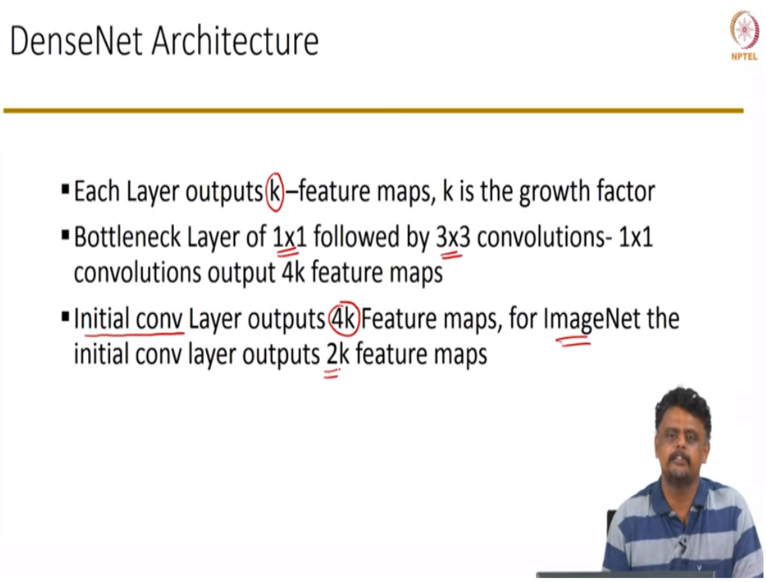
→ 1 Convolution layer.

G. Huang, Z. Liu, L. v. d. Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017, pp. 2261-2269.

So there are a few other terms that the paper talks about and which are the important concept as far as this dense nets are concerned which have this is summarized briefly in this slide so the growth rate this determines a number of feature maps output by into individual layers inside a dense block so in this case we saw that here we see about K equal to 5 for instance dense connectivity by dense connectivity we mean that within a dense block each layer gets as input feature maps from the previous layer where as seen in this figure shown in this figure and there are transition layers which transition layers aggregate the future maps from a dense block and reduce it is dimensions.

So max pooling is enabled so is one by one convolution, composites function so the sequence of operation inside a layer goes as follows so you have bytes normalization followed by an application of value and then a convolution that will be one convolution layer these are the operation that are done in a convolutions layer, so all these four concepts are basically the ones that underlay a ten percent architecture.

(Refer Slide Time: 07:03)



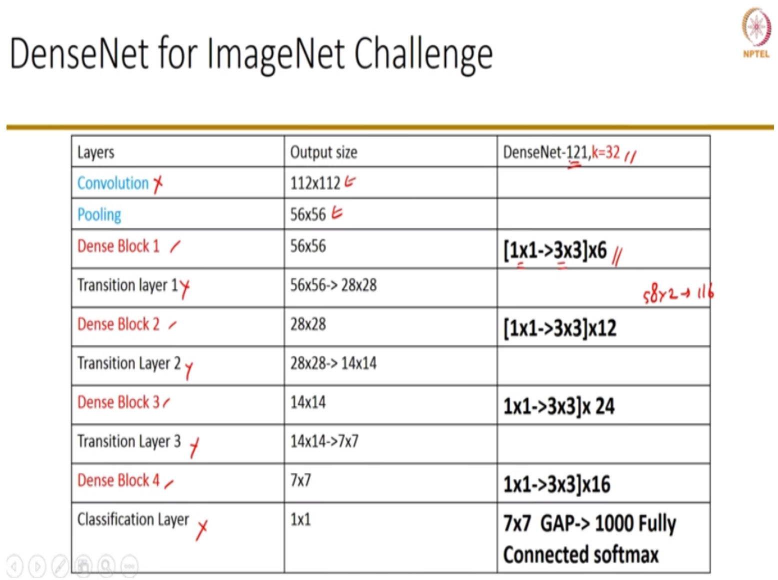
DenseNet Architecture

- Each Layer outputs k –feature maps, k is the growth factor
- Bottleneck Layer of 1x1 followed by 3x3 convolutions- 1x1 convolutions output 4k feature maps
- Initial conv Layer outputs 4k Feature maps, for ImageNet the initial conv layer outputs 2k feature maps

In general let us some basic details here so each layer outputs K if it is enough we saw that is the growth factor and as far the convolutions are concerned they also use this bottleneck concept which we saw in resonate as well as in inception so this is basically a one by one convolution followed by three by three contributions in general every one by one convolution outputs about 4K feature which are operated by the three by three contributions and before the input goes to a dense block there is an initial condylar which outputs about 4K feature maps and these 4K feature maps are then used as input to the first dense block and so on so forth.

Typically every network that they have designed for the image net challenge as well as other safe our database etcetera typically has about three five dense blocks and with a growth factor ranging from 24 to 32 and so on, as for the image net challenge the initial layer outputs about 2K feature maps.

(Refer Slide Time: 08:07)

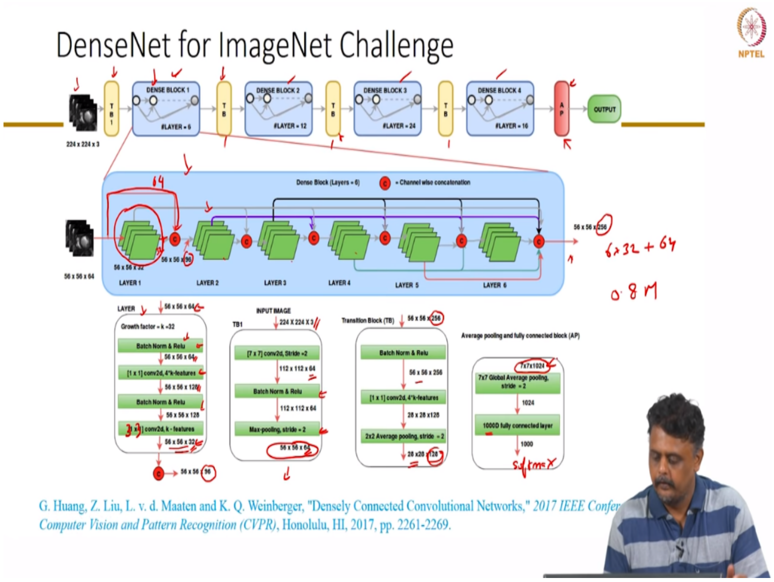


DenseNet for ImageNet Challenge

Layers	Output size	DenseNet-121,k=32
Convolution	112x112	
Pooling	56x56	
Dense Block 1	56x56	[1x1->3x3]x6
Transition layer 1	56x56-> 28x28	58x2 → 116
Dense Block 2	28x28	[1x1->3x3]x12
Transition Layer 2	28x28-> 14x14	
Dense Block 3	14x14	1x1->3x3]x 24
Transition Layer 3	14x14->7x7	
Dense Block 4	7x7	1x1->3x3]x16
Classification Layer	1x1	7x7 GAP-> 1000 Fully Connected softmax

So in this slide we look at this is a table which have summarized we look at one of the architectures but the instant it has 121 layer that they use for the image net challenge here the growth factor is 32 equal to 32 so initial convolution give rise to one twelve by on twelve feature maps followed by max pooling which about 56 by 56 and then there are about four dense blocks which define so the first dense block here defines six one by one convolution forward by three by three formation that is 12 convolution layers the second dense defines about 1one by one followed by three by three convolutions and 24 one by one followed by three by three India’s block three similarly for the dense block 4 you have 16 of these one by one followed by three by three so if you add these up so we will get some 6 times to 12 so from here I will get about 50 times two convolution layer so 116 of them and then we have three transition layers so that will be under 19 there an initial convolution layer about 120 and then the classification raised about 121 so that is why we have the incident 121 we will look at the details of each of these layers in the next line.

(Refer Slide Time: 09:26)



DenseNet for ImageNet Challenge

G. Huang, Z. Liu, L. v. d. Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks," *2017 IEEE Confer Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017, pp. 2261-2269.

So for the dense that here the images shown is that of a cardiac sequence but the Denson challenge used RGB images from the imaging database so if you look at the image net challenge network had about 4 dense blocks that so one we saw followed by and there are in three intermediary transition blocks as they called and there is an average global average pooling block when which connect to a thousand dimensional portal so what summarized here in this picture so let us look at each of these blocks so let us look at the first one TB one which is the transition block one or what we call as initial corner so the input image is 224 by 224 by 3 which is a typical standard crop used in my most algorithms in image net challenge now these are then subject to a seven by seven convolution with side of two giving rise to 64 feature maps that is to K because K equal to 32 for this particular architecture so K is the growth factor, the growth factor is 32 so you get about 64 feature maps for this particular network convolution then followed by batch norm and (())(10:41) and then a max pooling with the slide of 2 which give rise to a 56 by 6 times 64 feature map so 64 channel of sits 56 by 56 so now we look at this is the input to the first dense block here this is a layer block.

So now the first layer right here we were shown right here the word factor again is 32 as we saw earlier feed the first do batch norm followed by and relu which still retain the size of the feature maps to 64 physics by 56 and then we have one by one convolution which we resize to a feature maps which means that we have 128 feature maps of the same size these are conversions preserve the size I mean once again reactivation batch norm and relu and followed by a in this case it is this type here it is three by three feature map which gives rise to K feature or 32 features, so this is the output of the one of the convolution layers in dense block one the output of this after concatenation with the input will give rise to 96 feature maps because that is 32 feature maps here and the input is 64 feature maps so we add those two content those two using the rise to 96 feature maps, so this is for the one of the layer in the first dense block if you look at a transition block which is about is right say light right here this is one of the transition blocks it receives as input about 256 layers we can go through math the calculations and verify that it is indeed 256 so it takes these two fixed layers and again batch term relief followed by one by one convolutions which gives rise to 128 feature maps subsequently resulting in an output of 128 of size 28 by 28 because we do a two by two average with the side of to reducing the size of the feature map so this following dense block one after going through all the dense blocks when it approaches the average pooling block you have feature maps thousand twenty four feature maps of size seven by seven do global average pooling the side of two and then of course we get it is full connected to a thousand dimensional activation followed by soft max.

So this is the typical is one of the dense and dark texture used for imaginary challenge it has comparable performance to rest nets and other large network architectures that we have seen in the past but with a reduced number of parameters so far instance the one of the top performing dense and architectures had about 0.8 million parameters, so 800,000 parameters which is sometimes with order of three or four times lesser than some of the levels larger networks so with reduce turner we want but this is kind of counterintuitive because you are cannot get an instead of adding like in rest net you are concatenating features to subsequent layers however there are no new filters defined in every layer you control the number of filters in every layer by using a growth factor and we are using a small enough growth factor you will only define very few number of filters and subsequently a fewer number of parameters that has to be estimated.

So for a hundred layer in this case we saw a 121 layer network the number of parameters is of the order of hundreds of thousand now this has another benefit in the sense that it don not over fit so typically some for the large network you have hundreds of millions of parameters tendency to over fit and less dead augmentation and regularization is done so this kind of implicit temporary regularization it is also referred to as feature reuse because you are concatenating feature from earlier layers and using them and diffusing the filter kernels on top of them so that is another advantage and reduce number of parameters also help so this architecture is now we will see how this architecture can be further used as fully conditioned network for semantic documentation in the form of both encoder decoder network or units and see how their performance compares to other deep architectures thank you.

Let us look at one of the layers inside the first dense block just right here so it receives us input 56 by 56 by 64 feature map from the initial corner which he called TB1 here there is a batch storm and relu layer followed by one by one convolution which gives rise to 4K feature maps which is about 128 because K is 32 in this case and then of course again followed by a batch Norman Relu and in this case there is a mistake here this is three cross three convolution to produce K feature maps these K feature maps are all concatenated with the input feature maps so these are we saw that about 64 of them and there is 32 output so we will get about 96 feature maps which are given as input to the subsequent Conley inside a dense block now as you progress through each of these layers in the end we will have about so we have about one two three four five six each of them producing 32 feature maps plus your input feature maps of size 64 which will give rise to 256 feature maps which is the input to the transition block, the transition block of course does a sequence of one by one convolution forward by two by two average willing to give you 128 feature maps so this if you walk through this similarly for every other dense block and of course the transition block right here then we will end up with about 1024 feature maps of size 7 by 7.