



Machine Learning for Engineering and Science Applications
Professor Dr. Ganapathy Krishnamurthi
Department of Engineering Design
Indian Institute of Technology, Madras
CNN Architecture LeNet and AlexNet
Part 1

(Refer Slide Time: 00:14)



CNN Architectures

Instructors: Balaji Srinivasan & Ganapathy krishnamurthi

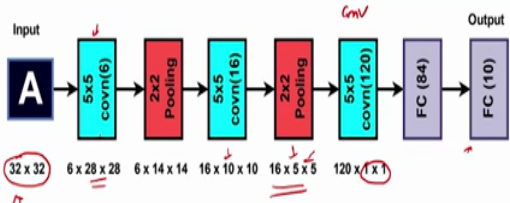


Hello and welcome back, in this video we will look at LeNet.

(Refer Slide Time: 00:18)

LeNet- Digit Classification


- Among the earliest CNNs to be used for digit recognition
- Based on multiple convolution and **average pooling** layers
- Number of Parameters **60k**



Average Pooling

→ Conv → Pool → FC → FC → Output

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86(11): 2278-2324, 1998



So Lynette 5 as it was called dubbed was first published reported back in 1998, it was one of the earliest instances of convolutional neural networks used for image recognition, it is

specific application was for digit recognition and apparently had commercial application where it was used to read millions of checks in banks. So this network serves as a kind of template for most of the most (ma) most of the modern networks that we see today.

So we will just briefly look at it and see some of the variant features of this network. So this network took as input what 32 cross 32 images these are in this network was trained with images from the ((01:00)) database, so the images were about the size 28 by 28 and then these images were then further modified to in fact kind of like a data augmentation when there is a several distortions were introduced and was for training this network.

So it took us input images of size 32 by 32, it is basically images of digits handwritten digits which has been discretized scanned and discretized. So the typical architecture is basically input followed by a convolution followed by a pooling layer and this was repeated leading to finally to a couple of fully connected layers and an output which is basically one of ten classifications you have to classify zero digits 0 to 9.

Network at about 60,000 parameters it has several interesting concepts here for instance the first layer had 5 by 5 convolution no zero padding which leads which gives rise to a 28 cross 28 output followed by a 2 by 2 pooling this is an average pooling operation so basically 2 by 2 average pooling with a stride of 2, so the average pooling basically the output is basically the average of the four elements in that 2 by 2 area of the filter then followed by a 5 by 5, 5 cross convolution again no padding which gives rise to a 10 by 10 output and then subsequently another 5 by 5 convolution gives and then a max pooling which gives rise to a 16 cross 5 cross 5 maps.

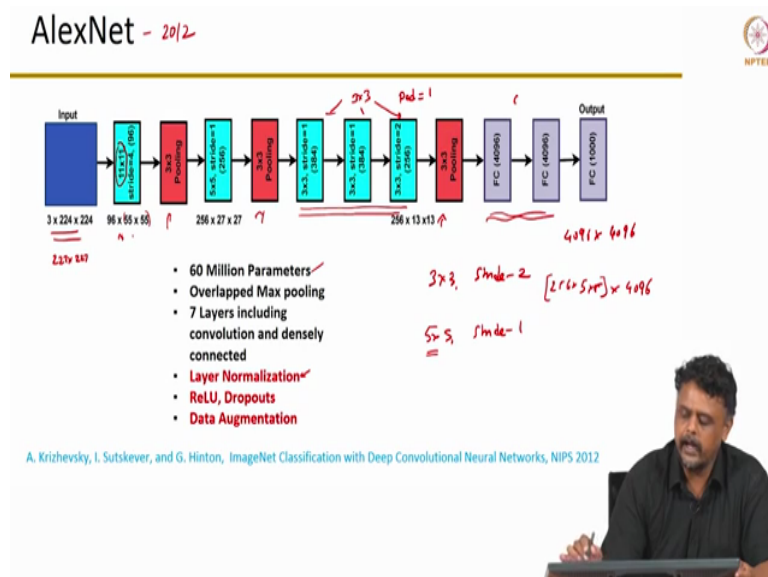
Now when we do a 5 cross 5 conviction on top of these maps it is basically the same as doing a fully connected layer but then the paper actually refer just call this a also calls this a convolution layer is also a convolution layer, so we craft we start with the 32 by 32 input here perform 5 cross 5 convolutions followed by max pooling and then one more 5 cross 5 convolution followed by a max pooling ok which gives rise to 5 cross 5 feature Maps 16 of them and we do again if 5 cross 5 convolutions on that give giving rise to 121 cross 1 outputs that is the interesting part, the author also mentioned ((03:30)) is the first author of this paper he mentions that this is what he need, right now we call this as fully convolution layers right.

So if we have an input which is bigger than 32 cross 32 then the number of feature maps in this the size of the feature maps in this layer would be higher than 1 cross 1 in fact if you think about let us say a 64 cross 64 input one way of looking at it is that we can we can stride the (net) entire network across the 64 cross 64 with a stride of 32 and we can actually get 4 2 cross 2 output here, so if you can think about it that way then finally a fully (convl) connected layer and then to a decision layer or the output layer we can use soft max here if you want ok.

So this has several things which are repeated even now in current networks one is that as you go deeper into the network starting from the input the size of the feature maps typically shrink because of the convolutions also by the pooling operation which is actually subsampling operation and not only that as the size of the feature maps it decrease the number of feature maps increases, so in the end we have 121 cross 1 feature maps and the first layer we have about 6 (cro) 6 featured maps of size 28 by 28.

So this general principle is you will see is reflected in along in the current architectures also, so this network achieve state of the art results in 1998 for digit technicians based on the (()) (04:59) digit database.

(Refer Slide Time: 05:02)



In this video we will look at AlexNet which was this work was done in 2012, so AlexNet was the entry to the image net large scale image recognition challenge and this was among the first CNNs to be you know entered into the challenge and it actually beat all it is a (compo) it is nearest competitor by more than 10 percentage points ok, so this was say deep neural network with about 7 layers ok and 60 million parameters ok, so this was the structure of this

network is the architecture of the network is very similar in terms of a very similar to LeNet 5 in terms of the convolution and max pooling operations but then it had various other innovations in it as we will see in this video.

So just to give a brief overview of this network so the imagine challenge the input these are RGB images so the input layer at size 224 cross 24 cross 3 some sources have pointed out that it is Act should be actually 227 cross 227 in order for the output in the second layer to be consistent ok, so the first layer first convolution layer had 11 cross 11 filter, 11 cross 11 filter with the stride of for giving rise to 96 feature maps of size 55 by 55 and then we have a max pooling operation here max pooling operation using a 3 cross 3 kernel and a stride of 2, so which needs to effectively having the size of your feature maps, again followed by 5 cross 5 convolutions.

So this is the typical structure so we have all max pooling operations are 3 cross 3 with the stride of 2 and all the convolution operations are again 5 cross 5 with a stride of 1 ok, so the layers you see here in the intermediate convolution layers you see here without any max fully they had convolution with the padding 1 to preserve the size of the network and in this case so I really mentioned that 5 cross 5 for almost all of the convolution operations but in these intermediate layers the filter kernel size was 3 cross 3 for all these 3 ok.

And they had a padding 1, so in these boxes the intermediate convolution layers are shown here has their convolution filter kernels of size 3 cross 3 with a pad of 1 to preserve the size of the feature maps forward by max polling and then a fully connected layers to full fully connected layers and an output which is one of thousand, so the image net classification challenge provides you thousand image categories and you have to classify them as one of thousand.

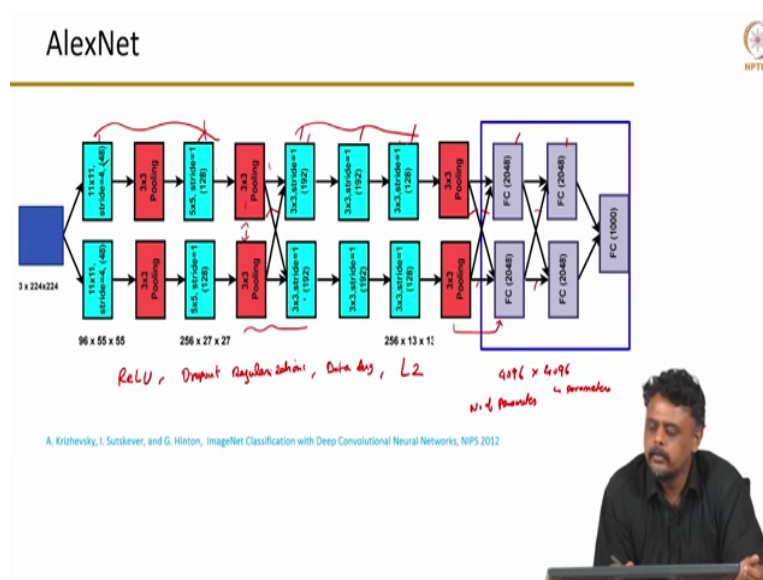
So the pretty deep network at about 7 layers so if you count it is about 1, 2 about 5 convolution layers about 3 max pooling ok typically about 7 layers which included convolution as well as the densely connected layer, so if you if you just count the convolution plus the fully connected layers you will have about seven of them, they this network also had a normalization layered layer normalization where it is called (coloca) local contrast normalization this was done by if you look at a particular pixel or an activation in a feature map you normalize it is value by looking at the adjacent feature maps at the corresponding feature location.

So this was but of course this is no longer done this is one this was the one off thing has done for this particular realization of the network. So another thing that this network again had you know if you look at this network lot of the computations happen in the earlier layers because you have 11 cross 11 convolutions 5 cross 5, 3 cross 3 convolutions ok but if you look at the fully connected layers in the end lots of parameters here most of the parameters come from here because you see the if you look at these two fully connected layers this is like 4096 times 4096 weights ok.

And if you look at the max pooling following this is what 256 cross 13 cross 13 as input to the max pooling layer with the stride of 2, so you will have about 256 cross in this case if you have pooling 3 by 3 you have took first 5 cross 5 I think and you unroll this and follow up with a fully connected layer to a 4096 activations ok. So lot of the parameters occur towards the end and lot of the computations are at the you know input side of the layer B of the network because of the size of the convolution columns.

So if you see this network actually has a mix of 11 cross 11, 5 cross 5, 3 cross 3 correlation kernels.

(Refer Slide Time: 10:19)



A. Krizhevsky, I. Sutskever, and G. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, NIPS 2012



VGG

- Developed by Simonyan and Zisserman for ILSVRC 2014 competition.
- VGGNet consists of 16 convolutional layers with only 3x3 kernels
- The design opted by the authors are similar to Alexnet i.e. increase the number of the features map or convolution as the depth of the network increases
- The network comprises of 138 million parameters.

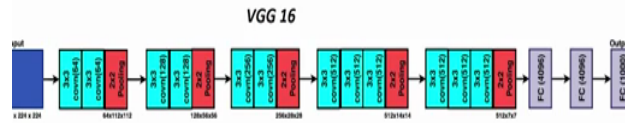


Image Courtesy: <https://upload.wikimedia.org/wikipedia/commons/c/c5/Vggnet.jpg>



So let us just look at this convolution layer at the input, so this one aspect of this network which is now highlighted in this figure is that the Gpus available at that time the one this particular network was trained had only 3 gigabytes of input of memory available, so the computations for this network were split across 2 GP, so if you look we have split the network from the previous stride into two pathways.

So basically the top here all the computation in the top half here would go to one Gpu and all the computations at the bottom of here will go to another Gpu ok, so for the first layer we have 96 filter kernels as we saw so 48 of them in 1 and 48 of them in the other, ok. So if you look at the number of parameters here it is very easy just to show an illustration now to calculate the number of parameters let us say on just one pathway we have each filter has 11 cross 11 parameters ok, so we have about 48 of them in one ok and there are two pathways in this case GP 1 again and GP 2 so $(11 \times 11 \times 48) \times 2$ parameters per convolution layer in one convolution layer, ok.

And it produces 96 of course feature maps so we have to that is why we have 48 times 296 which feature maps ok. So the size of the feature maps is 55 by 55 so these are a number of parameters, so if you want to calculate the number of computations that you have to do per Gpu for let us say number of computations per Gpu so you have the output size is 55 cross 55 times 48 those are the total number of activations in the on one Gpu in this particular computation layer convolution layer and for each of these we have to calculate the are this is a number of for each one to produce each output we have to do 11 cross 11 cross 3 computations, right.

So this is the number of outputs activations so basically 55 sorry 55 cross 55 cross 48 and for each output we need to perform 11 cross 11 cross 3 multiplications yeah I have you know the additions that we have to do because it is the sum of products you can actually you can put in a factor there for it if you want to but this is a typical way to calculate number of computations, of course the network is spread across two Gpus, so this is for one half of the feature maps so you have price as many okay.

So this is the typical computation number of computations that you it is how you would calculate in the network and the number of parameter you see the number of parameters is quite less actually in initial layers right and if you go to the final layer again I say short before in the earlier slides you can at least see for between the two fully connected layers you can calculate so it is 4096 times 4096 this is the number of parameters, so that is your weight matrix for a fully connected layer which is very (sim) it is how you would calculate for a regular artificial neural network, of course the number of parameters coming from the pooling layer to this layer is again is much higher because after the pooling you would have to unroll it and then do make it into a fully connected layer ok.

So this is typical computations like when so a lot of the number of parameters here are huge there is a number of parameters that I calculated ok but a lot of the computations happen in the late earlier stage also ok. So there is another innovation here I say you referred I said that as two pathways that is because if you look at how the training are spread across two Gpus you see that except for the inner layers where I have the cross arrows these are the layers where I have cross arrows in all the other layers the ones here and here the computations are restricted to the feature maps in that Gpu.

So this has so let us say if you look at this particular layer this has 48 feature maps and we do a 3 by 3 pooling but the pooling operation only draws from these feature maps and when you do the subsequent convolution with the 5 by 5 kernel then in the 5 by 5 kernel actually acts across only this region that is why we call it the multiple pathways or two pathways because and if you come here in this layer then the output activation here also has feature Maps includes feature maps from both the Gpus.

So this one layer we have computations are or involve both the G or are from feature maps in both the Gpus while in other layers they are done independently, so this is like having a separate pathway 2 different pathways in the network ok and this is one of the networks the

first time use of ReLU we saw that an earliest slide ReLU non linearity, it also used dropout regularization we saw that earlier dropout legalization it also had dead augmentation so data was augmented images were augmented by because you see the number of parameters is huge there are a few million training images but we have 60 million parameters we do need more data if you have so many weights.

It also used L2 regularization ok, so that it does not know over fit to prevent over fitting some extra so dropout and L2 where circularizers see the dead augmentation was done by a flipping shifts translations as well as jittering the rgb values so they have done dynamically on the fly as network was training ok. So the network was placed first in the image tracking recognition challenge especially in the top 5 and the top one category terror it had an error of about 15 percent which was much higher than the second place finisher again one of the earliest and the first CNN 2 win the image that competition.

So this network sparked a huge interest in deep learning so it had lots of parameters was quite deep seven layers and if you include the max pulling also, seven weight layers if you can call it that because it had 1, 2, 3, 4, 5, 6, 7, 7 weight layers speak off and of course interspersed with 3 max pooling layers ok and it had a mix of filter kernels 11 cross 11, 5 cross 5, 3 cross 3 kind of systematic and if you also look at the how the number of features increased as we go deeper in the network.

So the initial layers we had first layer we had 96 and then 256 here because I am adding from both of the Gpus 384, 384, 384 was preserved ok, so as you go deeper in the network the number of feature maps increased, the size of feature max shrunk but the number of the representations increased ok, so this is a typical architectural design that most networks you see one now ok.

So you would as you go deeper in the network they would definitely be decrease in the size unless you do 0 padded convolutions appropriate 0 padded convolutions but then you can always increase the number of feature Maps to improve your representation, so that is the general principle it was also see that in LeNet 5 but it was much smaller in size because of the size of the input images as well as the as the problem is concerned, so it was just to recognize small images ok and there are also constrained by the computational resources available at that time.

So this was the AlexNet architecture ok, so in summary so AlexNet was the first network to be CNN to be used in use for the image net classification challenge produce state of the art research at that time ok, so it was about accuracy of 15 percent had a huge number of parameters about 60 million of them most of the parameters towards the decision layer so basically the fully connected layers contributed to most of the parameters had the design principle was to use multiple size filters here 11 cross 11, 5 cross 5, 3 cross 3 and the number of feature maps increased as you go deeper into the network ok.

It had two pathways the authors also report that having this multiple pathways actually improved their results, so two pathways in the sense the computations were split across two Gpus but if you see that as we saw the combinations were not shared ok, so except in one layer one or two less especially here not one layer this layer here and towards the end to the fully connected layers that is where the computations was share, ok.

Other than that the you can call this as two separate pathways ok, the results were reported based on an assemble of about seven networks multiple networks more than 5 networks ok, so that lead to also an improvement about two or three percentage points so multiple and the layer normalization was also introduced here but this particular layer normalization here is no longer used but they did have a normalization layer in between the conditional layers just before the input the convolution layer ok.

So now we will look at in the next subsequent videos, we look at other architectures that performed that gave state of the art results on the image net database.