Modern Computer Vision

Prof. A.N. Rajagopalan

Department of Electrical Engineering

IIT Madras

Lecture-25

So, CNN architecture right, we will leave this lenient, we will start with you know AlexNet. Now, this is how an AlexNet looks like I mean I wish I had a picture where let us say I did not drawn all this but anyway. So first thing is right this is actually 227 okay, on the figure it is represented wrongly 224 it is actually 227 by actually 227 by 3 okay that is the input okay that is going that is you know AlexNet is a one that actually that actually came up first in 2012 caught the attention of all right of everybody that was the first deep network to actually break this barrier right. So till then people were struggling to get these numbers okay this is that you know image recognition image net challenge right of visual recognition. So that is like that you see 1 million images with 1000 object classes and all right and there you see a jump from 25.8 to 16.

4 that was a kind of a wake up call right for you know for this one deep networks and the guy his name is Alex Krizvisky. So this is called AlexNet it goes after the guy's name okay who actually made this architecture and what you see is that one of the first things that you should notice is that right so when somebody defines an architecture for a neural network right they will say con 1, max pool 1, let us say norm 1, com 2 and so on that means that there are all these operations which are happening that means there is an input then there is a convolutional layer you are calling it as a first convolutional layer then max pooling is probably happening the first time then this norm is something that was only specific to this network this called a response norm okay and I do not think people use that anymore but anyway at that time it was there then you like like second convolution layer second max pooling and second norm then convolution 3 so all this you can see here so for example right this is the input right and here is here is a convolutional layer which is that con 1 then you have a max pooling right which is here and then you have a norm which is also happening somewhere here then you have a contour here then you have a max pool 2 then here immediately after max pool 2 you have this response norm and response norm and max pooling do not involve any unknowns okay response norm is simply like doing that you know y is equal to x minus mu by mu by sigma kind of operation but like I said right people do not use it anymore and then con 3 right is here and then followed by con 4 and then there is also a con 5 right and after that there is a max pool 3 then there is no norm after that then there is a fully connected fc 6 fc 7 then fc 8 which is the output layer right and if you see the number of layers right the way they are counted is like for example okay this is 1 right and then con right okay so you see the numbers that they are only increasing with con people are not adding max pool and all as a layer that's what I meant when I said that layers are typically where computations happen okay now if you if you just want to want to see this in a kind of

a little more detail right this is how it looks so right here is the input 227 cost you see 227 cross 3 and in the con 1 layer what you want to do is you want to have 11 cross 11 filters that's what he had okay by the way with a stride of 4 pad 0 and he wanted 96 such filters which means what he wanted 96 feature maps to come out from the input with filters of size 11 cross 11 cross 11 cross 11 cross 3 right 11 cross because the input is 3 dimensional right it has 3 channels so which then means that which then means that if you do this convolution it's if you kind of kind of go back to that formula you have 227 minus the filter size filter size is 11 plus 2 times 0 padding did he 0 pad no so it is like 0 by stride what is the stride is 4 plus 1 right this is that formula no so that means output feature map should be 227 minus 7 is how much 216 by 4 is 54 plus 1 55 right so which is what you see on the left right you see that 55 cross 55 cross 96 because you use 96 box filters okay now that goes as input then you do a max pooling if a max pooling 3 by 3 filters right I mean you know at stride 2 so what does that mean so you so you are you say you are getting a feature map volume which is like 55 cross 55 cross 96 and on this right you are doing a max pooling right which is what is this maximum so it's like 3 cross 3 right so you take 55 and then minus 3 right which is the filter size and I said that max pool also the same thing minus 3 plus 0 padding there is no 0 padding right so plus 0 by stride which is 2 plus 1 so what is this 55 minus 3 is 52 26 plus 1 so that is 27 right so yeah so here you see right 27 cross 26 and 27 cross 27 and you see 96 right and also right remember I have one more thing that I forgot to tell is when you are doing max pooling right you just do it independently each for each channel when you don't do like block max pooling I mean but in saying is you got you got to see 96 channels right when you are doing max pooling you are only looking at this local guy and then and then and out from there you get one channel then max pooling on this you get one it's not along the depth okay so max pooling is like you know each independent channel that's why you get still 96 channels for each channel you have done a max pooling and you have kind of reduced the the size effectively to almost half now from 55 and the idea is that you may still be able to able to do whatever you want right and and the other thing is you are carrying the most important information forward and I mean that's another way to look at it I mean one is that you are able to reduce the size of your feature representation each is like a feature representation right I mean at every layer you are getting some feature map which is supposed to explain something about your input image and the feature representation you know can can keep reducing and but then maybe you may not be able to make it too small maybe you lose something along the way therefore you have to play a night you know a delicate balance in terms of how much you can go down to there are kind of things called auto encoders whose job is precisely this whose job is to actually give an input image produce it at the output okay the same image and then right and then there will be there will be a bottleneck layer in between so the bottleneck is like saying how much can you squeeze it right and then the idea is that you throw away this the other part right which is the decoder part because the idea is that you have image whose representation is so neat that means it can be so compact right so that's called the bottleneck so there people will try to try to pull it push it down push it down push it down to the extent that you can still reconstruct reasonably but then this bottleneck becomes really a small sort of a representation which means that that representation captures all the what do you say right all the you know invariance that you

want to that you want to capture from any may think of a face and if the face has various different variations but then you want to ignore all those variations and capture what is intrinsic to that face right that is what you should be capturing and that typically will lie in a very very kind of a low dimensional space the rest of it is all noise right the dimension is getting the manifold is getting bigger because you are using all kinds of other extrinsics really if you want to look at in fact it is very interesting if you look at if you were to plot all our faces right I mean right think about it I mean you know let's say I take a take a human face 64 cross 64 and if I think about think about a space that is like 64 square right in that dimension if you pick a point will it be a face I randomly pick a point in that 64 square dimensional space typically it will not be a you will have to try very hard to find where a face is if you really look at where the face is that will lie on a very neat manifold you know it will be like very small subspace right out there whereas you are representing it using 64 square dimension thinking that that is what you need but actually that is not what you need because if I just pick randomly I won't see a face at all it will all be noise right so that is the idea behind representation right so when you keep saying representation that is what you mean what is that what is that space right in which in that subspace in which this whole thing is lying this this particular whatever right if you are taking talking about faces then where is it lying right so similarly all this feature representation feature that we keep talking about is all about getting there right and sometimes it is very explicit sometimes it is for a task like this where you want to do a classification okay let me just go forward so this max pool right you understood then this norm one forget it I mean you know don't worry too much about it then again a 256 5 cross 5 filters at stride 1 and pad 2 okay until you come here now one of the things right that I wanted to wanted to observe is that this I think you can figure out now see look at this the unknowns right that you have to find out the weights right I mean good that it is already here if you actually computed how many weights okay you okay you need to find out right that that that now you know right now you know the box filter size you know how many filters you need so at the first layer it is about you know 35 k at the second layer it is about 307 k at the third layer 884 k that means you are this is not max pooling at all this is exactly where you have a con 1 con 2 kind of layers that is where you will find out all the filters then 663 k then 442 k that means here you are at like con 1 2 3 4 5 right so here it is con 5 somebody can actually check this out okay you can match these numbers but but look at the next one the moment you jump from here to the fully connected layer it is like you know million it is 37 million right and why how do you arrive at the 36 37 million it will be like 6 into 6 into 256 which you are going to flatten right into 4096 right I mean you will unwrap this no 6 into 6 into 256 unwrap it and then put it next to another layer it will like 4093 you multiply that with a calculator you will get a 37 million so you can see the parameters suddenly blow up right I mean that is the reason why an MLP is not such a great thing because your unknowns right till now if you add up all this it may be a few million that is all that is 37 million then the next one is 4096 into 4 that is another 16 million then 4 into 1 k right that is another 4 million right so you have already so an Alex net has typically 60 million unknowns that you have to find out so in those days it was big deal right imagine right I mean 19 what is there no no not 19 2012 right 2012 you are talking about 60 million unknowns to be found out you do not have the GPS that is why you see that there are there

are 2 tracks here by the way did you see did you notice it is not like 1 track right you see there is an upper arm that is running and there is a lower arm that is running this you would not see in other network that is because they did not have the see GPU power so they had to have you know so you see right there are 48 feature maps here and 48 feature maps being independently estimated there because there are 96 no it split similarly 256 will be split like 128 128 so they are like 2 parallel arms going but do not worry about what is the upper parallel arms just that they did not have the compute power so they had to split that split the task so all that I am saying is if you look at what the convolutional neural network really occupied that must be not not just 37 million rest of it is about 3 million what the what the fully connected layer that came at the end occupied was like 57 million out of 60 million right it is a big thing but then if you look at the number of neurons where the actual computation is going on that you will find that the CNN will occupy most of the company so the places right I mean where does where does the computation happen it happens inside a neuron right you compute all these weights at all but where is the actual computation going on that is actually happening inside a neuron so if you look at look at how many neurons are getting involved in a CNN right it will be like 55 cross 56 55 into say 96 that many neurons and similarly 27 into 256 that many neurons into of course into the into yeah no the number of feature maps is already there right 27 into 256 so that number if you find out right you will have these numbers at that I that I have kind of marked here I mean you can compute that I mean so it is easy to just do the multiplication but you look at the FC right it is very small 4000 4000 1000 so if you look at the parameters the unknowns MLP is the fully connected let us not call MLP fully connected layers right will kind of you know take on the I mean they are the ones that will incur the highest computation highest unknowns right that you have to estimate and you know well you know one can argue as to why do then use a fully connected layer right can you not do but the idea is that you know somewhere apparently that kind of aggregation is needed in order to be able to solve these tasks classification task towards the end that you need at least a few fully connected that is what has been again empirically observed that you need that kind of aggregation you know where you have to aggregate from everywhere that is around this locality and all this okay till a certain point then after that right you have to have this where where you know where you have to have every neuron look at look at you know every other thing in the previous layer all this empirical okay like I said that none of this is apparently happens in the brain okay our the way we do things the way we represent to some extent maybe a few of these things happen but nobody knows how we do things okay so just that right so anyway so yeah so the number of neurons is really the other way around compare the unknowns wise it is like the you know the fully connected layers have the fewest okay and I think you know so see so historical node trained on GTX 580 GPU with only 3 GB memory so I just wanted to cover this right so if you look at look at the rate this was apparently the first use of relu okay some of the important takeaways of an AlexNet this is the first use of relu norm layers like I said not common anymore heavy data augmentation what does data augmentation mean? So like modeling the existing data in some ways. In some ways exactly flipping, adding noise, rotating, shearing right whatever you want to do so all kinds of augmentation. Dropout right now you understand but dropout is only for the fully connected layer I am going to leave it

to you why you cannot do dropout in a CNN can you do dropout if is it useful okay batch so when I say dropout 0.

5 that is for the fully connected layers okay batch size 128 that you now understand right mini batch 128 images at a time SDD momentum you understand now 0.9 learning rate you know now reduced by 10 manually when the validation accuracy flattens out right plateaus then L2 weight decay this is at what is L2 weight decay regularization on the weights yeah norm of w square right that lambda or beta or whatever that is multiplying that then this ensemble okay do not worry I mean that is like different architecture is ensemble sort of a performance so that I think we can leave out.