Modern Computer Vision

Prof. A.N. Rajagopalan

Department of Electrical Engineering

IIT Madras

Lecture-22

So, this convolutional neural networks right they are there again I mean you know this again a neural network kind of family except that it unlike the other one that we saw till now which is an MLP right there are certain things that change I mean when you try to use a CNN. And CNNs have been really good I mean in terms of in terms of it what they have been able to achieve and especially you know in the right imaging domain. So for example, for example, right I mean they have been applied very very successfully for in fact you know you would have already seen this face, face finding, detection, recognition right I mean then object recognition classification which we already right on the right in the right at the start we told we told about this object recognition then object locating an object right of interest then you can even look at you know. So and then segmentation all these are all these are going to say classification problems image segmentation then you can think of regression problems also something like denoising which is which is which is like you know cleaning up an image you can have de-blurring which is like removing the blur in an image. Then there are even things that actually combine to combine multiple domains what is called image captioning. Captioning means right give I give you an image and then and then you need to give us give a sort of a suitable title for it you need to be able to say what that image contains right.

So that is like both text and so there you have natural language coming in because you have to express something in terms of words. So the output is not an image right in image captioning the output is no longer an image the output is a caption. So the caption is like a text so you have a visual feature which is the image and then you have to kind of say come up with some kind of a textual sentence rate which says what that what that image contains right. And therefore right so what happens is all of this is employed CNN and the main thing that separates it separates this from the MLP is this word what is called a convolution right.

So this convolution is something that that completely separates it from right from here from here say MLP right and why is this so why is such an important thing right. See the thing is first of all we are all familiar when you do signals and systems we are all familiar with the fact that convolution is such a sort of you know clean operation right. So for example right so let us take an 1D case right we will just do a motivation. So for

example I have let us say an impulse response and then I have an input whatever it is m x m H n minus m right. So when you when you do this right so then what do you do I mean so you take let us say H is some kind of an impulse response ok in this case let us take let us take a finite support right.

Then what do you do you normally take a take a H of n you have an you have an input which is which is X of n and you want to find out right if this input is applied to this impulse response what will be the output right that is what you do. And that can be expressed in terms of a convolution provided that is a system is linear as well as invariant with respect to time otherwise you cannot write the convolution operations and if and only if right. If you have convolution it implies LTI, LTI implies convolution. Now if you see right so what do you do you take a you take a impulse response then you actually flip it about the axis and then that is like and then afterwards you multiply. So if you want your Y of 0 what we do you will simply flip your H of n and then at that at that instant whatever you kind of see multiply X and X and H the flipped H and then add up all the values that is what you put as Y 0.

Then Y 1 is what you kind of shift your shift your H n by 1 to the right then again again do the product this you people know right you have all done signals in systems course right that is what you do. But then what do you think is really what do what what is it that that gives it so much structure out there because you know that right I mean when you have something like this I mean you know. See any any kind of a linear system so even if this was H of n comma m instead of n n minus m I could still write this as a vector you know output as you know a matrix multiplying what is it X. Okay I can always write this but then this H begins to acquire a certain property right when when you when you replace H of n comma m with H of n minus m. In general you can write you know H of n comma m that means as you keep as you keep shifting the location H itself keeps changing.

But in this case right you do not do that right you just shift the H I mean you do not change your H right. So what is it so so this H right when you have an LTI system this H has a particular structure what is that called this is not an any old matrix I mean it is a matrix with a certain structure what is called a toplet structure have you heard of that right. So this H has what is called what is called a toplet structure and depending upon how you how we express it like in terms of you know if you write linear convolution in terms of a circular convolution that means you get a 0 pad X and all then this H actually acquires you know even better structure what is called a circulant structure. Okay and then and then then the moment right you have things like that a circulant structure that is when a Fourier sort sort of right begins to begins to begins to you know begins to act right. I mean you know do you guys know that the matrix that diagonalizes this H if it is

circular it is actually a DFT matrix are you guys aware of that.

See for example I mean if you have a circular matrix so what is what is the circular matrix A, B, C, C, A, B, B, C, A right this is actually a circular matrix okay this is also this will also be you know toplets that means if you look at the main diagonal and the off diagonal entries they will all be the same okay and it does not it does so a toplet does not imply circulant but a circulant will always imply toplet and circulant always square. Now the question is right a matrix like that when you want to diagonalize it right I mean diagonalize is something they keep on encountering so and if you actually a DFT which you have learnt is the matrix that diagonalizes this that has this unique power to diagonalize a circular matrix okay and that is how that is how we talk about in signals and systems right we go into the Fourier domain and all that but now the point is when you when you talk about a 2D convolution now this is all about 1D convolution which we understand right very well. So the moment you go to actually 2D convolution now first of all right the main thing okay that that that strikes you here is like I told you right where is all this structure coming from I mean why is it that we have so much structure and all when you write up and otherwise that H will not have this structure if you did not have convolution you would have you will have none of this structure but you get that because of the fact that there are actually 2 things which you have enforced one is one is actually well I mean okay well 2D well the okay and right hold on hold on right I mean I will get us come to that point so prior to that let me let me talk about a 2D convolution. A 2D convolution is what is good goes along completely the same way except that I have an image right which I want to which I want to convolve with some with some filter right it could be it could be a 3 cross 3 it could be a 5 cross 5 we do not take even filters because then the origin is not clear okay that is why we always talk about odd filters 3 cross 3 5 cross 5 whatever it is that you want right and what do you do we exactly convolve the same way so only thing is right if I had a filter then in 1D I would have flipped about the origin because I had only 1 axis but in 2D I have actually 2 axis right. So if I have let us say you know something like this A B C D E F and then G H I right then the way you would actually flow so if this is your original H and if you had to flip it right then whichever way you go it does not matter you can either flip it vertically or horizontally if I just flip it you know about vertically then I will get like 6 F I C F I then B E H I will remain here and then A D G will go here then I actually flip this horizontally then I will get like then I will get you know I H G on top then F E D here and C B A here okay whichever way you flip whether you do horizontally first that will be the flip filter and that filter is what is what you would actually apply here just the same way that you do 1D right you have an image here on which you want to apply the filter therefore what will you do we will take this you know flip filter you get a center it here and then again a weighted average just as you do in 1D except that this is all done on a 2D grid correct I mean same thing you know whatever you do in 1D is the same

thing you are doing here except that it is now running on a running on a 2D grid and then then you need to suppose suppose right that is your center of the center also right this is called a kernel or this is called a filter whatever you want to call it okay or these are called the weights of your filter right and then and then then you get a shifted from here to here then the center comes here then you again and then all of this output goes into an output grid and you store these values right so whatever comes out of here you can as a right you know put that here then whatever comes out of this you put here and that is how you get your output of course you know you will worry what will happen at the border and all that we will not worry about now but let us say right this is how you run your convolution now such convolutions right that you do so typically right these kind of filters that you do convolution with I mean why would you want to even kind of convolve with an image I mean why would I why would I want to do that why do I want to convolve something with an image what would I get let us say can somebody tell me why would I even want to convolve I said how you convolve but why would I even want to convolve what will I get one thing is you can get edges out of an image right if you have a gradient kind of a filter I can get edges I can I can actually I can actually blur an image of course each will require a different kind of kernel right not all filters can do the same job so if you want an if you want an edge right then maybe okay you can think about think about a simple filter like this so right maybe you can have 1 1 1 minus 1 minus 1 minus 1 0 minus 1 0 0 0 something like this right okay this will give you give you an x gradient then maybe right you can have a y gradient so that way right if you want to look at edges in an image you may use something like this or if you wanted to do blurring right then what will mean is if I have 3 cross 3 then I can put all 1s inside and then and then you know I can do a 1 by 9 outside if it is 5 by 5 it will be 1 by 25 so that right that's how you do averaging you can have a different blur you can have a Gaussian kernel there instead of a uniform kernel this is this uniform right all weights are same sometimes right you want to give more importance to the middle guy and then and then you have a Gaussian kind of filter you can have all kinds of filters there right and these are all handcrafted so in the sense that well if I want if I want some kind of in in fact you can even remove the blur with a kernel under certain conditions because that's an inverse problem can't always do it but assume that you have even a kernel to remove the blur in an image right so all these are like the filtering operations which can be equivalently expressed in the Fourier domain also but the point is right these are all filters that we hand craft and then these are things that we know but what we really want is you know a network that can actually figure out what these filters ought to be and it should actually figure out where what kind of filter should come in come in play and we don't want to sit and tell what should be that filter.

Now now now right I mean so so when I kind of say do this spatial convolution so so normally what happens is right these filters are very small in size if you look at a CNN

right these will this will not typically exceed a 7 cross 7 size your image may be very big your image can be you know 1024 by 1024 it doesn't matter but typically these filters are not really right that kind of big and all they I mean because whatever they can do they can do with a with a limited support. Now the the kind of key thing to notice right when you when you are doing this kind of a spatial convolution are actually you know are actually 2 things one is what is called actually locality one is what is called locality and another is what is called the spatial or let's say weight sharing let's call this weight sharing. What does that mean? So right locality means so for example right so I mean if you if you had an MLP right I mean you know you would have had suppose suppose I am looking at 1 neuron and I am trying to see what all it is seeing at the input it will see every one of them right and then you have weights for all of them but now in this case right in this case I mean you know if you see if you see that you know if you see if you kind of think of the think of the images as being a previous output or that's the input to your network and you are kind of looking at what should what should this neuron be seeing this neuron is actually seeing a very very local region. So just looking at some 7 cross 7 or some 5 cross 5 depending on the filter it's not seeing the whole image at all right and this actually people have found that you know especially for images this is also to a reasonable extent true for 1D for example even for audio signals right it's not true that you know there is a correlation across our entire time window. Similarly in an image what you generally find is that there is a lot of continuity locally see for example look around you know if you see locally right things will look very similar right I mean if you look at this desk I mean all the intensities around it will all be very similar but of course things will change as you move but then very locally if you see that things are very very similar right.

So one of the arguments that is being made is that don't have to look you know don't have to look everywhere and see this has changed over the years okay this notion is how it was built but later people felt that you know even looking elsewhere might be actually useful and that is where attention came right. So people started talking about attention and so on but that's something right that if time permits we will talk about but I know let's just keep it very simple in the sense that if I just if I just look at the local interaction if I capture the local interaction that is enough and there is something called a receptive field. A receptive field that means right what does a neuron see how much of an image does a neuron see right you have one layer you have second layer or third layer fourth layer and so on and let's say each time you have a small kernel this filter it's still small but if you really see effectively what this guy will see is much more than just than just a 3 cross 3 because this guy comes from a 3 cross 3 and each of these guys is seeing a 3 cross 3 neighborhood here and each of these guys is seeing another 3 cross 3 effectively what this end neuron will see is actually a fairly bigger part of the image but then indirectly but not directly right. The filter do you guys see this I mean as you go deeper

and deeper if you look at the receptive field is what receptive field is like how much of the image does one neuron see. So right now we are saying it's very local but it's local if you if you consider the first layer it's only is 3 cross 3.

Let me ask you right if I if I go to the second layer and if it is a 3 cross 3 filter how much of an area will will that will that neuron in the second layer see of the image? No how much will it see it's not square 3 cross 3 and then the next guy will also have a 3 cross 3 kernel right and therefore how much effectively of the input is that guy seeing? 5 cross 5. 5 cross 5 why because see right I mean you guys should should see it like this 1 2 3 4 2 3 4. Now if you have if you have right one guy one neuron here so neuron it will see will see like kind of a cone. So it will start from here and this cone right will be like this right so it will it will actually try to see see this area right that's all it sees but then here if you see if you see a neighboring neuron see right this neuron sees this what is what is this neighboring neuron see? This I mean right neighboring neuron will see will see an area like that right that is what the neighboring neuron will see is it not? A guy here will see like this left portion right with center here ok that is what goes there. And therefore, it a neuron next which is actually taking taking a cone like this what will that see? That will effectively see what each guy each of these guys has already seen of the input image that will be 5 cross 5 right you see that? See what this guy will see it looks like it is seeing 3 cross 3 right of the previous layer, but then effectively with respect to the input it is seeing 5 cross 5 it is not square ok.

So, the point is so receptive field will actually increase so it is not true that you are doing it very locally that is for argument sake, but actually if you have deeper layers then the end neuron will actually see a much larger picture of the of the image, but through all these interactions right it is not direct. The other thing is weight sharing right so when you are doing the convolution right what you are saying is you know all the the the the filter weights right are not at all changing right. So, when I when I go to an adjacent neuron what did we do in MLP right when we changed all the weights right when I went to the next neuron I had a fresh bunch of weights, but here we are not doing that we are just sharing the weights and right this is something that I will show later through an animation as to what this means, but for the time being right these are the these are the 2 key things sort of say takeaways are that you have only local interactions right which you are allowing and the second thing is you have weight sharing right that is what is going on. It is actually convolution is exactly that convolution we do not call it weight sharing at all because in convolution we say that the we say that if you if you what do you say I mean you know if you if you shift an input and the output by let us say tau the output will also shift exactly by tau that is that is exactly is what is what is what the impulse response does right, but instead of calling it impulse response at all we just call it a filter because we do not because there is nonlinearity and also we do not talk in terms

of impulse response at all. We simply say it is a filter whose weights are shared and and and a filter that is really local in terms of its spatial support.

You can also argue for all of this in terms of 1D, but because we are already into mostly CNNs is for 2D. So we will just we will just go to the go to the image domain right. So now let me just show you a picture of you know one very famous kind of a network right so that you know so that you you you get an idea.