Modern Computer Vision

Prof. A.N. Rajagopalan

Department of Electrical Engineering

IIT Madras

Lecture-23

Okay, so for example, right, so if you look at a modern CNN right, this is how, this was actually, this is meant for this a character recognition. Okay, this is how, this is how it looks. So for example, here is an input which is, which is an image, okay, that is going in. And then similar to your MLP, you have got, you have got various layers. But then in between a few things are happening which is right now not clear. For example, there is something called sub-sampling that is going on here.

We have not seen what that is. But there is, there is a convolution, right, which is going on here, some 5 cross 5 convolution. That means that, that actually filter is 5 cross 5, right, that is what it means. And then here there is another 5 cross 5 convolution.

And then there is again a sub-sampling which we have been talked about. And then, then suddenly right, there is an FC layer connected, fully sort of, you know, fully connected layer. And then finally, right, you have, you have a classifier. Now one of the things, right, that we should remember is that just because you have a 2D CNN, right, it does not mean that, mean that, you know, fully connected layers would not occur at all, okay, inside that. You can have a network that is completely, completely sort of convolutional.

Those are called FCNs, fully, fully convolutional networks, okay. So there is a, they are like fully convolutional, end to end, like a regression problem will typically be fully convolutional. But here that is not the case. Here your, your job is to actually, you know, identify, kind of recognize the input number, right. So you got like 0 to 9 and therefore, right, you have these class labels at the output.

So it is a cross entropy, right, at the output. And eventually, right, you have to, you have to have, similar to that one hot vector that I talked about, when you show a 0, let us say your output vector should be like 1 and then all other 0s, right, you have a 10 dimensional vector at the output. If I show you 1, then maybe you just say that the second digit should be on and everything else should be, should be off at the output and that is how you do a cross entropy. This we saw last time, right, and there will be a softmax and all somewhere, okay, that and all will be there. Now one of the things,

right, which is not, which is not clear is this thing, okay.

You seem to have, you seem to have multiple things stacked, right. One convolution I can understand, right, I mean, if I have to, if I have to do a convolution by weight sharing and locality, I can just take a kernel, do this, I can take a 5 by 5, but then you see that, right, there are, there are, there are multiple things stacked there. So, so the idea is that, right, these are, this is, these are called channels, okay, these are called channels and the idea is that, right, the idea is that each channel, okay, so for example, each channel is like, is like an output, correct. So it is like I have applied a convolution, whatever I get at the output is that channel and that channel is actually, there is another name for it, what is called a feature map, okay, it is called a feature map. So each channel, so the blue is one feature map, the one, one back to it, that is a white thick colored thing, right, this is a feature map, that is a feature map and just because both are drawn in blue does not mean that they are the same feature maps and all, just some, some sort of coloring, right, that they have given.

So each is actually a different feature map and, and, and right, if you are really wondering about why, why do we do that, okay, the answer is that, right, I mean, when you, like I said, okay, when you, when you do the initial layers, right, you want to, you want to capture what are called edge relate, what is typically edge related information and you might have one channel whose filters are such that, whose filters are such that, right, you know, you get all the, all the, one kind of edges, it could be simply horizontal. You may have, you may have another filter that can capture all the, all the, all the kind of vertical edges. You may have another, what you call, filter that may capture diagonal edges. So each channel, right, that you can think of is you can kind of think of, you know, think of something like that wherein the network figures out, again we are not telling the network, right, we are not telling that we need horizontal edges, we need vertical edges, but typically what will happen is the initial layers will try to capture edge information, which is like low level information, there is nothing very high level there, it is completely low level and that low level information will come in terms of multiple features which you actually extract from the, from the input image and each filter will try to, will try to be something like the one that I showed at 111 minus 1 minus 1, it does not have to be that. But very lightly that it takes a form, some of them will take a form like that, something could be even things that you cannot explain so very clearly.

So all these, so, so what will happen is, right, so what will happen is you, so one of the hyperparameters that you have is how many feature maps you want, right, that is again a hyperparameter, that is something that, that, that you have to tell, the network does not know. Suppose I tell that I need 8 feature maps, that means you need how many filters now? You need 8 filters, right, that means this network has to figure out what are those 8

filters such that when it applies, so one filter it applies on the image it gets the first blue feature map, then it finds out another filter, applies on the input image, gets a second feature map, then it figures out another filter, applies it on the input image you get, so you get 8 feature maps. Ok, so till this point is there, is there any sort of doubt I mean, so, so the idea is that, right, you can have filters of a smaller size but then each filter you give the power to sort of, to sort of, to sort of extract, right, as much as you want from the input image. And what is typically found is that the, the lower level, the, you know, the, the deep network behaves in a manner such that low level at the, at the, at the, at the, what we call at the initial layers, right, you end up actually, you know, it ends up figuring out what the, I think I may have some figure for that, right, so what these edge, so I think right here it is. So, so this low level filter, right, that they are called, so low level in the sense that these are the, you know, in the initial layers and you, and you find the kind of things, right, which they, which they actually find, ok.

And this is Gabor filter, ok, Gabor is just as I showed, right, a Gaussian filter, similarly a Gabor is, you know, a sinusoidal modulation over a Gaussian and this is an, this is an analytical filter which again is meant to capture orientations. And what has been found is even let us say inside our own system, ok, human system also we know that, you know, Gabor like filters exist in these initial layers, what is called, what is called the, what is called the V1, V1 layer, we do not know much about V2, V3 and all, but in V1 we know that roughly that is what happens in the initial layers and turns out that, that, you know, Gabor like functioning is what these filters also do, even though they may not be strictly Gabor filters if you try to look at the weights, but functionally this is what they do and each filter may capture a different orientation and so on. So, for example, right, I mean this is one filter, another filter, another filter, they are all stacked, ok. So, you can have sometimes 256 channels, ok, does not mean that, it does not mean that you can only 8 channels, need only 8 channels and all, you can have a, you can have as many channels as you want, but only thing is they should be useful, you do not want to just pile on channels, right. Now, what will happen here now, ok, suppose, suppose we ignore this layer, right and I want to, and I want to, and I want to go here, right, I want to go to my next layer and suppose I declare that I want, I want to see 24 filters, 24 feature maps, let us say, I want 24 feature maps in my second layer.

My first layer I said I have 8 feature maps, right, in my second layer I say that I want actually 24 feature maps. Now, again, again I need filters, ok, now what will I do? But here I have got 8 feature maps already, right, that is the input now, right, the 8 feature map is what is going as input to my next layer, not x, ok, x was at the input, that was the image, we acted on it, we got something out. So, we have 8, 8 out, what do you call, 8 channels, right, 8 feature maps, they are all going as input to the next layer, right. Now, what kind of a filter will you use now? So, what you use is a box filter, ok. So, what is a

box filter? So, box filter, right, will look like this.

So, box filter will look like this. So, what you have is, right, one channel, then behind this is another, there is another, there is another and you want to aggregate information from all of them, right. I mean it is like, it is like one input, no, I want to aggregate information. See this could have happened even at the input, it does not mean that at the input you get only 2D. For example, if it is a color image you will have RGB, you will have actually 3 channels, not one channel, ok.

So, it is not like at the input you will have only a 2D layer, even at the input you can have 3 channels, but I thought to start with you will keep it easy. But as you go inside it, now what you need is a filter that, let me just show it by a different color, that will run like this, right. That will be the, that will be the box filter. So, suppose I say that my size is 5 cross 5, that means, that, so that will be the spatial extent, but the extent in terms of the depth that it goes, that will be exactly the number of channels that I had in the, from the previous output. So, that I do not have to tell, that will be automatically 8, because I have to span all of them.

So, I cannot leave anything out, because I got the whole input to kind of look at. Do you guys follow this? So, it will be like a box filter now. So, it is box filter, but the weights from channel to channel will change, ok. It does not mean that, it does not mean that the weights, suppose it is 5 cross 5, it does not mean that this 5 cross 5 is the same as what is being applied on the, you know, next channel.

This will all vary. So, it is eventually, right. So, if I have, if I have let us say 3 cross 3 filter and I have, and I have 8, the 8 such channels that means I have these many unknowns, ok and right, this is not enough, because in the next layer I am asking for, I am asking for how many did I say 24. I am saying that I need to see 24 filters, right. Now one box filter is for one channel. I need another box filter for the next channel.

I need another box filter for the next channel. I have got 24 like that. So, you need, yeah, exactly. So you need into actually 24. I mean that many weights you will need and plus, so until now I only talked about weights.

So plus the bias. The bias as you can clearly see is one, is the, is one number for one channel. Just as the weights are all constant for one channel, right. Similarly the bias is a constant for the whole channel. Therefore it will be, so, so all these weights will come, right. So when you add these weights and then, then when, when, when this neuron takes that weighted average it has to, you have to also add a bias to it, right.

So, so that bias will be actually 24. So you will have, you will have 24 channels, right. You have got 24 outputs. So you need 24 bias there and you need 3 into 3 into 8 to cover one box filter and you have got 24 outputs. This is like 24 box filters plus the bias which is like 24 biases.

Is this, this is clear because this, this I want you to grasp in your head and one of the things that, that, that you might wonder is why do not we translate. After all, right, it looks like a 3D box. Now why do not I go like, see I am going like this, right. I am doing a 2D convolution. I mean I am translating this now.

See how, I mean how will I, how will I get this first feature map? I will have to use this box, apply it at that location that will only give me one number. Then I have to translate this, no? I have to move it across the whole box, right, I have to cover. But that will give me only a 2D output. That would not give me 3D because I am summing, I am summing along the, along the depth, okay. So you can actually think about each one as a convolution of that filter with a respective channel and just add up all the outputs, right.

Either way it is the same. So I will get one value. Then I translate, I will get one more value. That is how, that is how I will arrive at this, at this feature map, okay. And then I have to change my filter, then I get the second feature map, then I, that is how I get my 24 feature maps. But the point is that you might ask, well, I am translating horizontally, why am I not going depth wise? Why am I not going this way, right? Actually you can also do that.

Strictly speaking that is called, that is a pure sort of a 3D convolution. But this is all, the filter is 3D but then the convolution is still actually 2D because you are not letting the filter translate along Z, okay. I mean there are actually reasons for it. I mean, you know, in special cases we do but in, but most of the cases we don't do that, okay.

We don't translate along Z, okay. So that is something that you should remember just in case you are wondering why don't we get that. So that is why you get only a 2D map, you don't get a 3D map. If you did that you would have already gotten a 3D map then. I mean if I also translate along Z, I won't get a, I won't get a kind of 2D, right. I will get actually a 3D map, 3D feature.

But I get actually a 2D feature and I kind of do this, repeat this with multiple box filters so that I get, I get, right, that many feature maps. Then if you see this figure, right, so all of this you can continue doing and so you can go from here to here and you can actually find out how many what will, so the size and all we will see later. If I apply a 5 cross 5 what will happen to the size of the feature map? That is something that we will see later.

But for the time being you should just understand if somebody gives you an architecture, right, you should be able to figure out what is going on there, okay. So from here to here you can come and then eventually, right, there is, there is, I mean if you see here, right, I mean you have sort of reached a 1D, 1D thing, okay, right.

So what, so that is called flattening. So what this simply means is if I have an image I will simply, simply, you know, right, unfold it into a single sort of, you know, one-dimensional, one-dimensional sort of, you know, quantity. So this is also something else that you can do what is called, well anyway, I mean for the time being, right, just, just, just think of an image which is getting unwrapped, right. I mean even if you have multiple channels it does not mean just one channel. You know multiple channels you just unwrap it. Whichever way you want to unwrap because then the network will figure out accordingly, right, it will try to find out what should be the weights.

See you need that because eventually you are trying to solve a classification problem, right. So you need, you need a one-dimensional layer at the output. So this flattening will happen and after this flattening now it is MLP after that, right. All there is no, there is no more weight sharing, it is fully connected. Then then you come down, you decide how many layers you want after that, then eventually the last layer should have only a 10 label.

So that is clear. But in between it how many layers you want that again is something that is a hyper parameter nobody knows. After flattening, right, should you go like 2 more layers before you actually converge to that, you know, 10 label sort of output layer or, you know, should you be doing something else that is all hyper parameters nobody knows. So each network will have something, okay. But, but what I want you to take home is the fact that, right, there is a CNN, right, depending upon the problem at hand whether there is regression or classification it can have a connected layer also, fully connected layer towards the end, okay.

That is when it happens. Initially and all it will all be, you know, 2 dimensional filters. But after that it will all be, so, so here the number of parameters will actually blow up here. You know that, right, because it is, it is fully connected. I mean here, here you might be frugal, you know, right, in this region, right, you would have been very frugal because, I mean, you just have 8 feature maps, right, the 3 cross 3 filters. So if you look at the number of unknowns it is hardly anything.

You are local, you have shared the weights. So there is really number of unknowns really very, very small. But the moment you go to a fully connected layer, right, I mean you see that things are blowing up. Of course, you know, here the one way the weights

blow up is, you know, depending upon the number of feature maps that you are asking. The more feature maps you ask then the filter size will also become that much bigger and therefore you will actually increase your unknowns but then, you know, it may still be worth it because it is still local, right. So this locality and weight sharing is what is, what is kind of, you know, what is most important for a CNN and yeah, okay.

And then you can have, you know, a connected layer which is, which is, you know, you can have a fully sort of, you know, connected layer also.