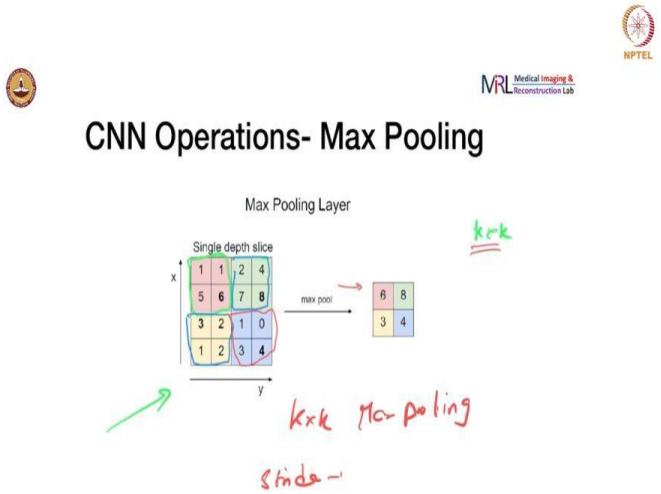
Medical Image Analysis
Professor. Ganapathy Krishnamurthy
Department of Engineering Design
Indian Institute of Technology, Madras
Lecture 44

Max Pooling

Hello and welcome back. So, in this video we are going to look at the one additional operation apart from convolution, that is done in a CNN; Convolutional Neural Network. So, we looked at convolutions and strided convolution, different types of convolutions. Now, we are going to look at the one other operation called max pooling, which also forms an integral part of CNN. It is not that widely used anymore; but can be used for down sampling at least; I will see that in the next few slides.

(Refer Slide Time: 00:48)



So, if we have a feature map as shown here, a feature map as shown here, and this is basically one feature map from a sequence of feature maps. And if you want to down sample it, in order for that you get a slightly sub-sampled version of it; then, you can do what is called the max pooling, where you choose n×n or k×k regions of the input, and choose the maximum value in there.

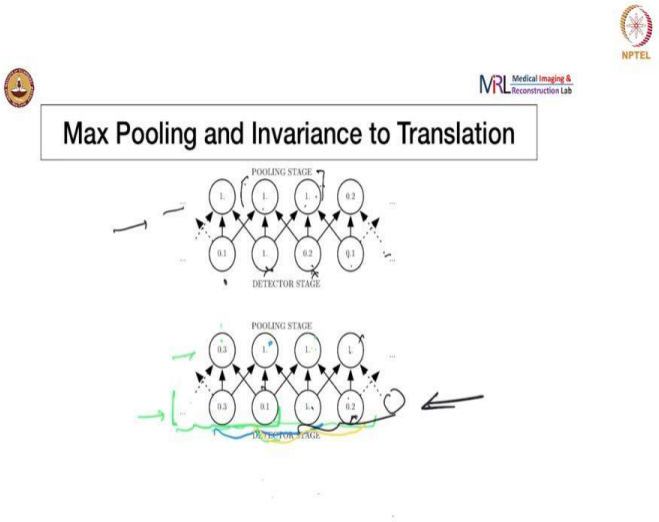
So, as you can see from go from left to right see, see this this high region I am I have drawn; I highlighted the maximum value 6, and that goes to the output. Here, you have another region,

you have a 8 as output; and in this region, you have 3 and another region you have 4 as the output the output. Because, they correspond to the maximum inside that two cross (region). So, you usually do this as a k×k max pooling. So which, which basically means that you look at a k ×k neighborhood and find the maximum value in that neighbor.

And you can also do that with a stride. That means that you can; so in this case, it is done with the stride of stride of two. That is how you get the reduction in what you called the sub-sampling effect, reduction the size of the feature map. If you can also do it at the stride of one, where you will not get the as much as dropping the resolution; but you will have some other purpose.

So basically, what this does is make the network slightly translational invariant; we will see how that works. So, first purpose, if you do with a stride, it helps definitely helps in reducing the size of network; so you can treat it like a filter. The other one is that it makes to some extent the network translation invariant at least two smaller translations; so we will see how that works in the next.

(Refer Slide Time: 02:42)



So, we look at how max pooling helps with translational invariance to a certain degree by looking at it from a typical, the typical 1D kind of input not the 2D; further we have for images. So, we look at the image at the bottom first, so just two stages; one is. Before we do max pooling, this is the output; this is called the detection stage.

And so the from the detector stage, basically that is the output from let us say convolution; and we are going to do this max pooling. So, if you look at his max pooling state, so, so for instance, this output; this is the output of max pooling, it takes as input from three neurons here or three hidden active units here; and so it turns out be one is not shown.

So, let us say the output is point 3 that is the maximum; this max moving it is 1×3, max pooling. So, if you look at these three, if you look at this particular hidden unit, which is the output of which is the pooling stage; then this takes its input from like these three units, so the output maximum is 1.

Similarly, if you consider this unit, this takes its input from combination of different ideas. So here these three, so that is also the maximum again is one; similarly, for the last one. For here, consider three, there is one here which is not shown here. Once again, the maximum was one, assuming that the 1 that is not shown is smaller than 1.

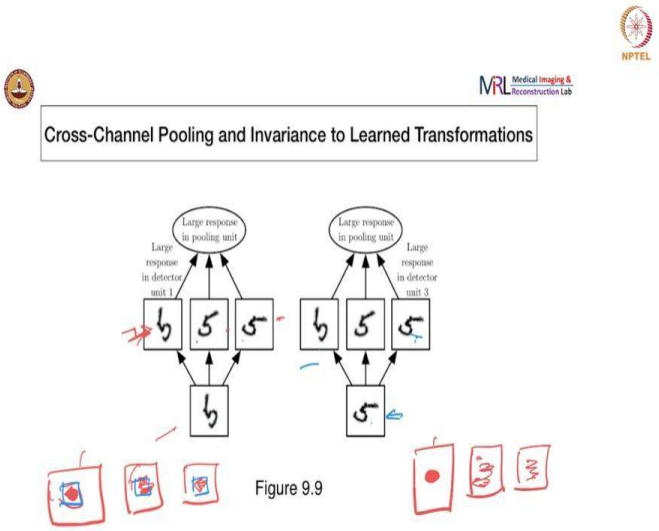
So, if you see we are just doing 1×3 max pooling. In this case, the stride of 1; we are not skipping anything, so the resolution is maintained. On the other hand, if you look at the picture on top where we have shifted the inputs to the pooling layer to the left, so this particular layer here has been shifted to the left by one unit.

So, which means that again once again, there is something here which I think is 0.1; say as 0.1 and come in here, 0.2 moves here. So, this moved there, 1 moved here; say the 0.1 moved there and this moved outside. So, now once you look at this output and you do the pooling. If you do the pooling you see that still for these two especially outputs remain the same. So, this is what I meant by invariance to translation. So, regardless of the shift, these two outputs still the maximum in this.

So this layer, these activations did not change, they still remain 1; for if you if you apply max pooling that remains. So, if you look at the bottom, almost all of the units have changed their value, and compared to the bottom here after shifting; but the pooling in the pooling layer, only two values have changed. So, the basically 50 percent of the values become unchanged. So with small translations, the output is unchanged; so that really helps. There is another way to do pooling which is across channels. And so that means that for every filter, there is an output.

And if you do max pooling across different outputs of different features, then there are some invariants that we can get for rotations for instance. So, that can also be solved, so we will look at that.

(Refer Slide Time: 06:16)



Let us see how we can do max pooling across feature maps. So, the previous version we saw is the max pooling only in in a layer in that in the for a for a 2D convolution neural network. It is it is it is basically a feature map; you are looking at 2×2 regions. The 1D example we saw, we looked at 1×3 regions. So, if we if we consider let us say, some task like where you are trying to do digit recognition; then we will consider the network which has learned three filters. So, these are the filters, it has learned. And for this for the sake of argument, these are the same size as the input.

Let us say we give this input image 5 or the number 5; then you know that this particular filter here, it is going to have the maximum response. So, each of them will give a feature map, the feature map output; I am just going to draw very crudely here. So, the leftmost feature map will have some. If we looked at like a cross correlation, this will have very large activation here.

Others might not be so large everywhere, there might be some partial overlap, especially at these corners; but they will be very small, something like this. I am just going to shade them, so that it is it is kind of all over the place; but this has a very large activation here. And if we do max

pooling across, let us see max pooling across this feature maps over small regions. Then, we still have a very maximum activation corresponding to the numeral file.

Similarly, if we look at these three, if you look at this particular diagram here; see we have input which is oriented in a different way, then this filter will give maximum output. So, if you draw these filters, I know like let us, it can be just 1.2; but I am just going to draw it like. So, you have very maximum activation here. The either two might have spread out low intensity activation because of the partial overlap.

But once again, the activation due to numeral file will be very high here. So, this particular activation map can then be used for recognition, digit recognition and downstream layers. So, if you do this max pooling across layers, of course, you have to decide what is the size across which you want to do pooling etcetera; but, those are hyper parameters that you have to study. So, even in this context, cross channel pooling in this case depending on the filters learned; it gives you invariance rotations, so it is minor rotations. Not large rotations, but minor rotation is also depending on what the network has learned.