

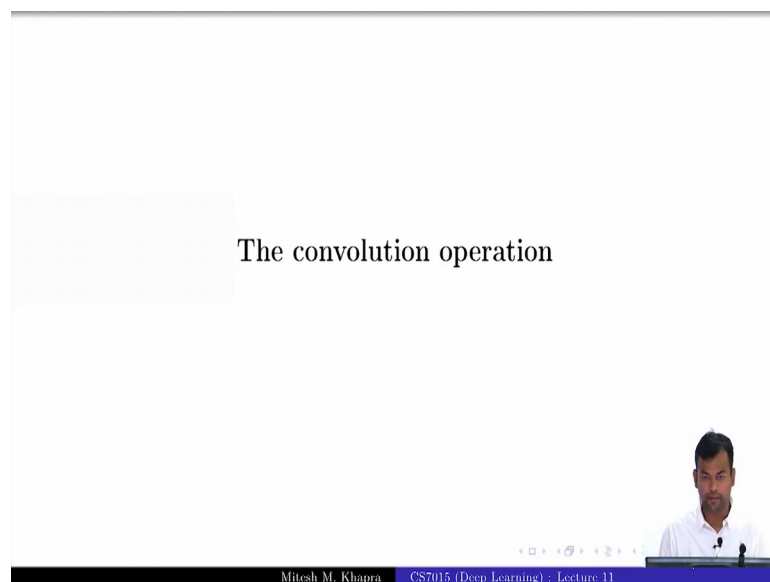
Deep Learning
Prof. Mitesh M. Khapra
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 11
The convolution operation

So, why let us start. So, far in the course we have looked at feed forward neural networks, we have seen how to train them and we have seen 2 special cases of feed forward neural networks, one was the auto encoders for learning, representations or learning latent representations of inputs and the other thing that we had seen was how to use a feed forward neural network to learn word representations where we saw this word to vector algorithm and its different variants, it was continuous bag of words, skip gram model graph and so on. So, those are all since some specific applications of the feed forward neural network.

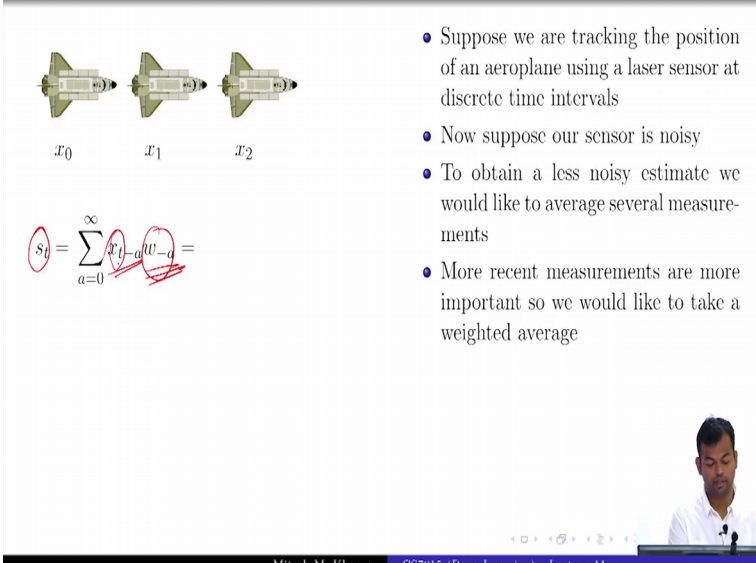
And now we will move on from there, though we will look at different type of neural network today, which is convolutional neural networks and we look at some specific architectures, which have become popular over the past few years.

(Refer Slide Time: 01:07)



So with that, I will start this lecture on convolutional neural networks. So, in the first module we will look at the convolution operation ok.

(Refer Slide Time: 01:11)



The slide displays three airplane icons moving from left to right, labeled x_0 , x_1 , and x_2 . Below them is the formula for a weighted average:
$$s_t = \sum_{a=0}^{\infty} x_{t-a} w_{-a} =$$
 The terms x_{t-a} and w_{-a} in the formula are circled in red. To the right of the formula is a list of three bullet points. At the bottom right of the slide is a small video inset of a man speaking. At the bottom of the slide is a black bar with white text: "Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 11".

- Suppose we are tracking the position of an aeroplane using a laser sensor at discrete time intervals
- Now suppose our sensor is noisy
- To obtain a less noisy estimate we would like to average several measurements
- More recent measurements are more important so we would like to take a weighted average

So, let us see, so, suppose we are tracking the position of an aeroplane using a laser sensor at discrete time intervals right. So, you have this ok. So, you have this aeroplane, suppose it is going from say Chennai to Delhi and at discrete time intervals, you are seeing the tracking the position of the aeroplane right, how far it is from Chennai at this point right may be it is 50 kilometers, 100 kilometers and so on.

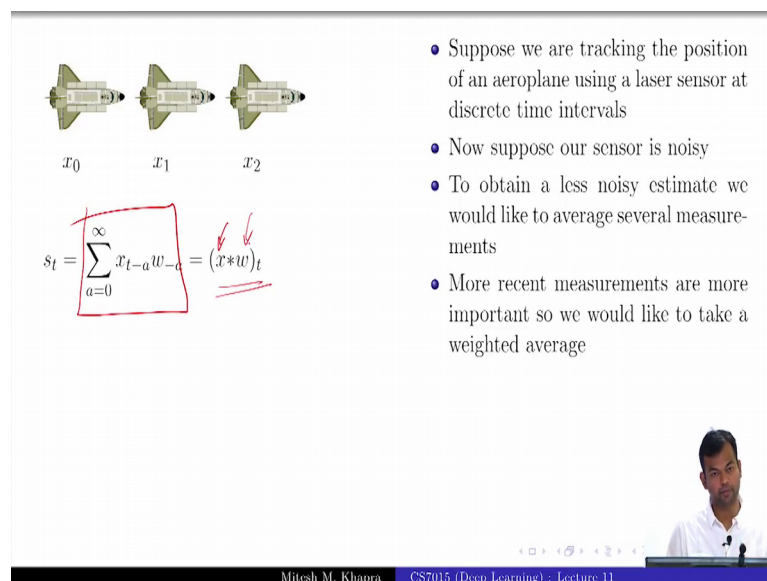
And now your laser you think that it might be noisy, it might not be giving you very accurate measurements. So, you would be taking these measurements and say intervals of course, it is not in practice you would not do that, but just indulge me for the purpose of illustration that say, you are taking these measurements every 5 seconds or 10 seconds or something like that.

Now, since your sensor is noisy instead of relying on a single measurement, you would probably want to take the average of the past few measurements that you have taken. So, that would give you a more accurate representation of what the current position is. Does that make sense like, you are taking multiple measurements and taking averages of those right and of course, more recent measurements are more important as compared to the previous measurements right. So, this is suppose at time step t , say this was t minus 5 seconds and this was t minus 5 minutes suppose.

So obviously, you would not want to take give a very large weightage to the measurement that you are taken t minus 5 minutes back right because, the plane would

have moved by a lot by that time. So, it rely more on the recent measurements and less on the previous measurements right. So now, the mathematical way of lighting, this is that you know the positions or the readings that you have taken at time steps 1, 2, 3, up to time step t, you are interested in the revise estimation of this measurement right. So, you have taken some measurement at time step t and you want a revised measurement of that and the way you are going to compute, that is you are going to take a weighted average. So, w is the weight of all the previous measurements that you have taken right.

(Refer Slide Time: 03:17)



x_0 x_1 x_2

$$s_t = \sum_{a=0}^{\infty} x_{t-a} w_{-a} = (x * w)_t$$

- Suppose we are tracking the position of an aeroplane using a laser sensor at discrete time intervals
- Now suppose our sensor is noisy
- To obtain a less noisy estimate we would like to average several measurements
- More recent measurements are more important so we would like to take a weighted average

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 11

So, the measurement that you take a t minus 1, t minus 2, t minus 3 all the way up to t minus infinity and for each of them would have a weight associated with this.

So, this operation right this thing, you can write it as the following operation that you have a vector of measurements or an array of measurements, which is x and you have an array of weights associated with these measurements, the farther the measurement from the current time step hopefully smaller is the weight assigned to that and this operation is known as the convolution operation right.

(Refer Slide Time: 03:37)

$$s_t = \sum_{a=0}^{\infty} x_{t-a} w_{-a} = (x*w)_t$$

input convolution filter

- Suppose we are tracking the position of an aeroplane using a laser sensor at discrete time intervals
- Now suppose our sensor is noisy
- To obtain a less noisy estimate we would like to average several measurements
- More recent measurements are more important so we would like to take a weighted average

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 11

So, you have x which is the input, w is known as the filter and the operation that is defined as this equation is known as a convolution operation right.

(Refer Slide Time: 03:49)

$$s_t = \sum_{a=0}^6 x_{t-a} w_{-a}$$

	w_{-6}	w_{-5}	w_{-4}	w_{-3}	w_{-2}	w_{-1}	w_0	
W	0.01	0.01	0.02	0.02	0.04	0.1	0.5	

X	1.00	1.10	1.20	1.40	1.70	1.80	1.90	2.10	2.20	2.40	2.50	2.70
---	------	------	------	------	------	------	------	------	------	------	------	------

S						0.80						
---	--	--	--	--	--	------	--	--	--	--	--	--

s_t

$$s_6 = x_6 w_0 + x_5 w_{-1} + x_4 w_{-2} + x_3 w_{-3} + x_2 w_{-4} + x_1 w_{-5} + x_0 w_{-6}$$

- In practice, we would only sum over a small window
- The weight array (w) is known as the filter
- We just slide the filter over the input and compute the value of s_t based on a window around x_t

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 11

Look, but of course, in practice you would not do this from infinity right, you would probably keep a window, you will say I will rely on the previous 6 measurements; that means, whatever I took at t minus 1 second, t minus 2 second up to t minus 6 seconds right, beyond that it does not really make sense. So let us see, how this computation

happens. So this weight array so now, what would be the dimension of this weight array? How many entries would it have?

Student: 7.

7 right 0 to 6. So, 7 entries and this is what my situation looks like right. So, this is the X the measurements, which I have taken using the laser ok. So, I have taken some measurements. Now, I am at a particular time step and I want to make a revised estimate. So, I have this x_t and from that I want to compute s_t and the way, I am going to do that is by taking a weighted average of all these previous measurements. Is the setup clear to everyone, ok?

And now this is what my formula is going to be. So, the revised estimate of s_6 is going to be whatever was x_6 into w_0 ; that means, the weight assigned to the current time step; x_5 into w_{-1} ; that means, weight assigned to the time step minus 1; x_4 into minus 2 and so on. So, you get this ok. So, I have these 7 weights and I will multiply with them with the 7 previous readings, 1 is to 1 multiplication and I will get the weighted average and using that I get a revised estimate.

(Refer Slide Time: 05:29)

$$s_t = \sum_{a=0}^6 x_{t-a} w_{-a}$$


W	w_{-6}	w_{-5}	w_{-4}	w_{-3}	w_{-2}	w_{-1}	w_0
	0.01	0.01	0.02	0.02	0.04	0.1	0.5

X	1.00	1.10	1.20	1.40	1.70	1.80	1.90	2.10	2.20	2.40	2.50	2.70
---	------	------	------	------	------	------	------	------	------	------	------	------

S	1.80	1.96	2.11	2.16	2.28	2.42
---	------	------	------	------	------	------

$s_6 = x_6 w_0 + x_5 w_{-1} + x_4 w_{-2} + x_3 w_{-3} + x_2 w_{-4} + x_1 w_{-5} + x_0 w_{-6}$

- In practice, we would only sum over a small window
- The weight array (w) is known as the filter
- We just slide the filter over the input and compute the value of s_t based on a window around x_t
- Here the input (and the kernel) is one dimensional
- Can we use a convolutional operation on a 2D input also?



Mitesh M. Khapra
CS7015 (Deep Learning) : Lecture 11

Now, I want to get a revised estimate for the next entry, how will I get it? I will just slide this weight matrix right.

So, I will just slide it by 1, I will again do the same computation and get the revised measurements. Again for the next entry, I will slide it by 1, slide it by 1, slide it by 1 and I will keep getting these entries ok. So, everyone gets the setup, how do you do the convolution operation? It is basically, a weighted average of the previous entries fine.

So, here the input as well as the kernel is kind of 1 dimensional right you. So, you have it is. So, you do not have a 2 D input here, you just have a single dimensional input here.

(Refer Slide Time: 06:01)

- We can think of images as 2D inputs
- We would now like to use a 2D filter (mask)
- First let us see what the 2D formula looks like

$$S_{ij} = (I * K)_{ij} = \sum_{a=0}^{m-1} \sum_{b=0}^{n-1} I_{i-a, j-b} K_{a,b}$$

$I_{33} K_{1,1}$

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 11

Can you use a convolution operation and a 2 D input also? Do you know of any 2 D inputs? Images, right? So, we can think of images as 2 D inputs. Now again I am trying to do the same thing, the setup is the same, the story just changes from laser to a camera now. So, I have taken an image maybe the image was captured and I am not very confident about all the pixels that I have captured ok.

So, now for any given pixel, I want to re-estimate it, using its neighborhood that is what I want to do ok. So, this is the pixel, I am going to look at some neighborhood around it right. So, every cell here is 1 pixel, just assume that every cell here is 1 pixel. So now, I am going to re-estimate this pixel by taking a weighted average of all its neighborhoods right. So now, can you tell me, what is my filter going to look like in this particular case? My filter would be just 3 cross 3, right. So, whatever neighbors I want to average on for every neighbor, I want a weight. So, if I am going to average on a neighborhood of 3

cross 3 then for each of these, I will want a weight. So, my filter would also be of size 3 cross 3, how many of you get this? Ok.

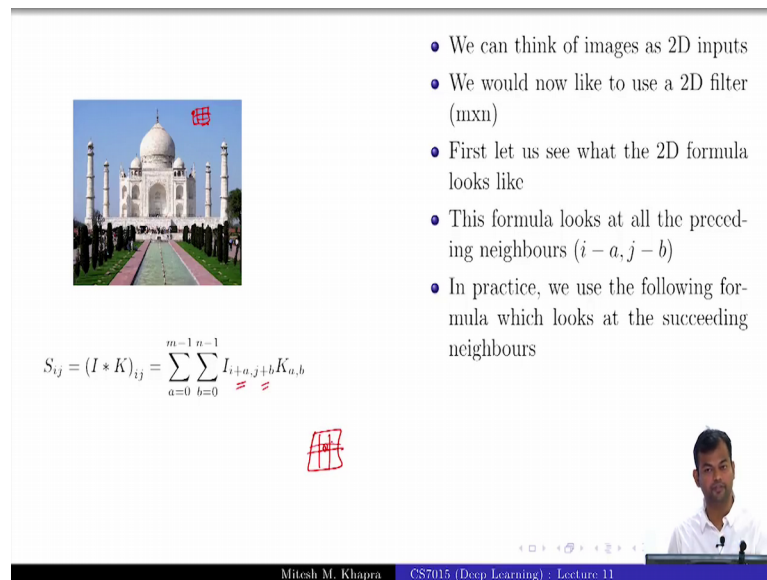
So, we now like to use a 2 D filter, which would be m cross n and in general it would be m cross m. So, it would always be a square filter, but I am just taking the case. Now what this nasty looking formula is doing, right? So, I have a particular pixel. So, this is an image. So, I will refer to this pixel as I_{ij} right. So, it is the i th j th entry in the image, I want a revised estimate for that I want an S_{ij} for that.

So the way, I am going to do that is I am going to look at m rows and n columns before it right. So, I am going to look at this neighborhood of m cross n and for each of these, I would have a weight associated with it. So, if I am looking at say for example, this was 4 comma 4, this pixel was 4 comma 4 then, I will look at 4 minus 1 comma 4 minus 1. So, that would be $I_{3,3}$. So, I will look at that neighbor and with that neighbor, I would have some weight associated, do you get that how this formula expands?

So, this formula would have m cross n terms, for every term, you would have a have a weight and that weight, you can just represented as this filter matrix. So, you get this what this formula is doing? It looks a bit nasty, but it is just the weighted average of all the neighborhood that you have and the neighborhood is a 2 dimensional neighborhood, in this case. How many if you get this properly, ok?

Now this, in this formula actually, I am looking at minus a and minus b; that means, I am looking at previous neighbors right. Now you should have these questions right, why previous neighbors, why not future neighbors? So, why am I not looking at this neighborhood?

(Refer Slide Time: 08:51)



- We can think of images as 2D inputs
- We would now like to use a 2D filter (mxn)
- First let us see what the 2D formula looks like
- This formula looks at all the preceding neighbours ($i - a, j - b$)
- In practice, we use the following formula which looks at the succeeding neighbours

$$S_{ij} = (I * K)_{ij} = \sum_{a=0}^{m-1} \sum_{b=0}^{n-1} I_{i+a, j+b} K_{a,b}$$

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 11

So there is no correct answer here, different convolution operations, I mean different packages use different convolution operations, but the most standard one, I believe is when you look at the next neighborhood right; that means, you had at this pixel and you will look at this neighborhood, the neighborhood after it right not the before it ok.

And in fact, so this is the formula that, I am going to look at plus j and plus p; that means, I am looking at pixels in the rows after this and in the columns after this pixel, all of you get this instead of before? Now what is even more natural to do? The names surrounding thing right. So, I will have this pixel and I will look at it is such a way that, this pixel is the center of the neighborhood right. So that is what I am going to go towards after a couple of slides and that is what, I will use for all my convolution operations, but in terms of textbook definitions these are the definitions that, you will find in textbooks ok.

(Refer Slide Time: 09:35)

Input

a	b	c	d
e	f	g	h
i	j	k	l

Kernel

w	x
y	z

Output

aw+bx+ey+fz			

m=n=2

- Let us apply this idea to a toy example and see the results

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 11

So, let us let us apply this to a toy example. So, I have this input, which is 2 dimensional input, I have a kernel which is a 2 cross 2 kernel. So, my m is equal to n is equal to 2. So, I am going to place this kernel at this location and then what will I get as the output?

(Refer Slide Time: 10:01)

Input

a	b	c	d
e	f	g	h
i	j	k	l

Kernel

w	x
y	z

Output

aw+bx+ey+fz	bw+cx+fy+gz	cw+dx+gy+hz
ew+fx+iy+jz	fw+gx+jy+kz	gw+hx+ky+lz

- Let us apply this idea to a toy example and see the results

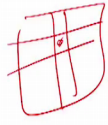
Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 11

a into w plus b into x plus e into y plus f into z right and I will keep sliding this to get the other entries. Do you observe something about the input and the output?



Student: (Refer Time: 10:07).


Size, the output size has reduced, why? We will get back to this.

(Refer Slide Time: 10:19)

$$S_{ij} = (I * K)_{ij} = \sum_{a = \lfloor -\frac{m}{2} \rfloor}^{\lfloor \frac{m}{2} \rfloor} \sum_{b = \lfloor -\frac{n}{2} \rfloor}^{\lfloor \frac{n}{2} \rfloor} I_{i-a, j-b} K_{\frac{m}{2} + a, \frac{n}{2} + b}$$


- For the rest of the discussion we will use the following formula for convolution



Mitesh M. Khapra
CS7015 (Deep Learning) : Lecture 11

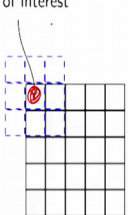
So right now, I just want you to notice it is obvious nothing great about it, but I will just get back to it more formally later on.

So, for the rest of the discussion, we will use the following formula for convolution, which is the centered formula right. So, m by 2 to m by 2; that means, I will be looking at a neighborhood, which is centered on the pixel of interest that is why this minus m by 2 to plus m by 2. Is that fine?



(Refer Slide Time: 10:35)


$$S_{ij} = (I * K)_{ij} = \sum_{a = \lfloor -\frac{m}{2} \rfloor}^{\lfloor \frac{m}{2} \rfloor} \sum_{b = \lfloor -\frac{n}{2} \rfloor}^{\lfloor \frac{n}{2} \rfloor} I_{i-a, j-b} K_{\frac{m}{2} + a, \frac{n}{2} + b}$$

pixel of interest



- For the rest of the discussion we will use the following formula for convolution
- In other words we will assume that the kernel is centered on the pixel of interest



Mitesh M. Khapra
CS7015 (Deep Learning) : Lecture 11

So, this is how I am going to look at it. So, this is how I will place, if this is the pixel of interest, which I want to re estimate, I will replace the kernel such that it, this pixel lies at the center of the kernel ok.

(Refer Slide Time: 10:49)

$$S_{ij} = (I * K)_{ij} = \sum_{a=-\lfloor \frac{m}{2} \rfloor}^{\lfloor \frac{m}{2} \rfloor} \sum_{b=-\lfloor \frac{n}{2} \rfloor}^{\lfloor \frac{n}{2} \rfloor} I_{i-a, j-b} K_{\frac{m}{2}+a, \frac{n}{2}+b}$$

pixel of interest

- For the rest of the discussion we will use the following formula for convolution
- In other words we will assume that the kernel is centered on the pixel of interest
- So we will be looking at both preceding and succeeding neighbors

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 11

So, we will be looking at both preceding and succeeding neighbors ok.

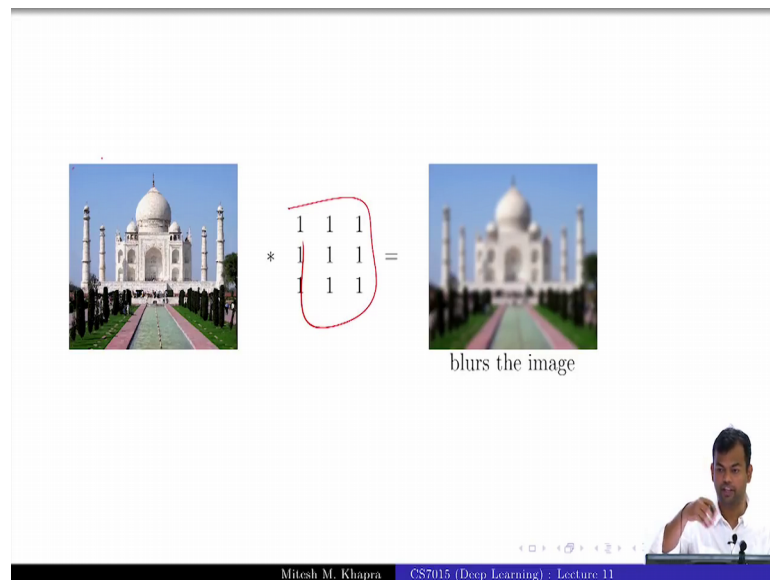
(Refer Slide Time: 10:51)

Let us see some examples of 2D convolutions applied to images

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 11

So, let us see some examples of 2 D convolutions applied to images.

(Refer Slide Time: 10:57)



So this is an image, I decide to apply the following convolution operation to edge ok, tell me what the resulting image would be?

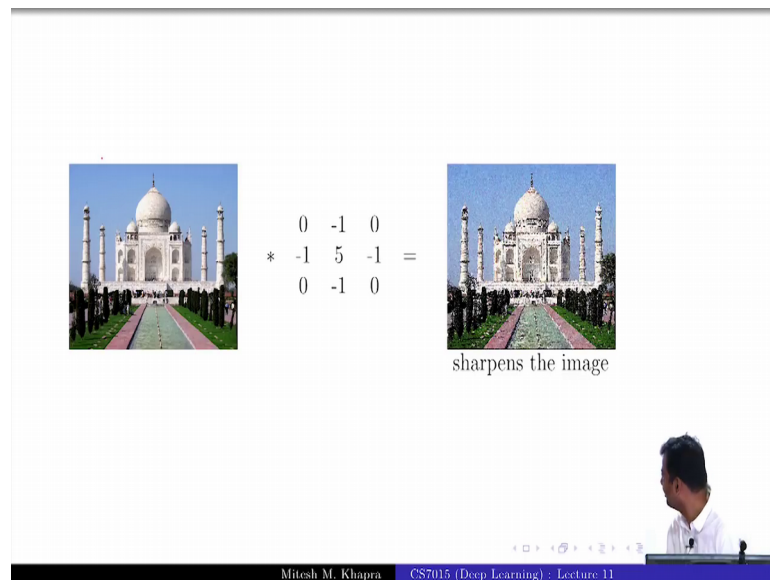
Student: Blurred.

Blurred, why blurred?

Student: We are taking average.

We are taking the average right. So, it would be blurred, you get the intuition. So this kernel basically, I have fitted at every pixel and I have computed the average around it and place at pixel by that average value and when we are going to take average things are going to get blurred right, because all the sharpness is gone ok.

(Refer Slide Time: 11:23)



Now let us look at this kernel, what will this do? Sharpen why? Because one was blurred the other has to be sharpened, what is happening here?

Student: (Refer Time: 11:29).

It is subtracting the neighbor's right. So you are taking 5 times the current pixel and subtracting the neighbors from it right. So, if the neighbors are similar, those would get subtracted and this would stand out really right, does that make sense?

This will result in a, but this in my on my laptop, this looks like a sharpened image, I do not know why it is looking like this here ok, it is a sharpened image just trust me, you can so actually are common right. So, people who have used adobe or any of these photo shopping software's right. So, you have this click button and where you say take an image sharpen and blur it. So, this is exactly what the tool is doing in the background, it is applying this convolution operation throughout the image.

So, when you say blur it is basically, placing that convolution operation throughout the image and computing the blurred image and same for sharpening and all these other spatial effects that, you have most of them come out of some convolution operation ok.

(Refer Slide Time: 12:23)

detects the edges

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 11

So for example, the next one, what would this do?

Student: (Refer Time: 12:26).

So, I will give you a hint, when will this result in a 0 output?

Student: (Refer Time: 12:32).

When all the neighbors are the same as this, right so then, when will it result in a nonzero output?

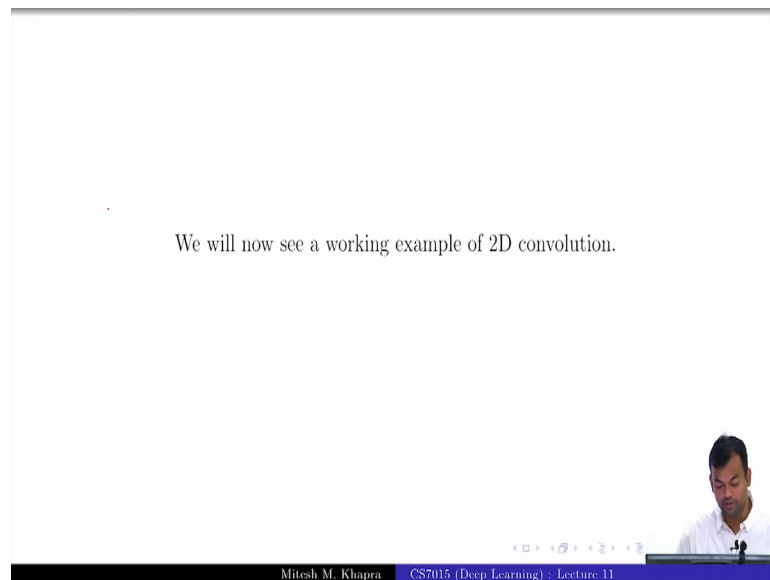
Student: (Refer Time: 12:37).

When there is a difference, when there is a difference. So, looking at this image tell me one place, where you know that it will result in nonzero output?

Student: (Refer Time: 12:45).

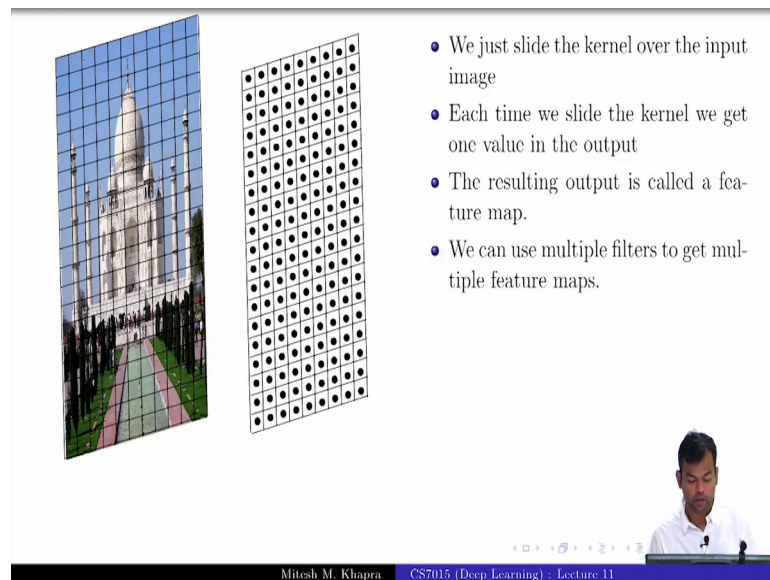
All the boundaries right. So, this is basically an edge detector in the slides, it appears properly ok. So, this is basically an edge detector and you get the intuition that these boundaries, whether neighbors are not the same as the current pixel, you will not get a 0 value. In this case, when all the neighbors are the same as the current pixel, so you are taking the sum of the 8 neighbors and subtracting the current value 8 times. So, that would be 0 right ok.

(Refer Slide Time: 13:11)



So enough of examples, so now, we will see a working example of a 2 D convolution. So, I just want to drill this idea of what happens, when you do a 2 D convolution.

(Refer Slide Time: 13:19)



So, what we are going to do is, we have this 3 cross 3 kernel and assume that everything here is a pixel ok, everything here is a pixel. So, I am going to slide this 3 cross 3 kernel across this filter, now when I place the filter once on the image how many outputs do I get?

Student: 1.

1 output. So, if I keep sliding it across the image, I will keep getting 1 1 pixel in the output ok. So, what the resulting thing that I get is known as a feature map ok, because it is the original input that we have taken. For every pixel, you have tried to approximate it or whatever filter weights you have applied and it necessarily does not mean, that you are taking an average, it could be some weird average of your neighborhood right. So, you have extracted some features from there.

So for example, in the edge detector case, you could think of it that you have extracted the feature that this pixel does not lie at a boundary right, that is why you get the black pixel. Do you get that? You see this way of interpreting a convolution operation that, you are trying to extract some features from that neighborhood.

So, in this earlier example whenever you got a black you are basically extracting the feature that this pixel does not lie at a boundary. Is that ok, fine? So now, you could get one such feature map by using a single 3 cross 3 convolution operation ok. If I use multiple such convolution operations, what would happen? I will get multiple feature maps ok. So, let us try to understand this, what is the dimension of my original image, m cross n into 3. Why is it into 3?

Student: RGB channels.

RGB channels ok. RGB is what we will have right. So, we will have this 3 cross m cross n. So, we will return back to this idea and from now this one image by using a single kernel, so this, in fact in for this figure right, I am assuming that the input is 1 cross m cross n and I am not assuming there are 3 channels although, it is a colored image, but just bear with me. So, it is a 1 cross m cross n image and when I apply a filter, I get a 1 feature map, if I apply k such filters, I will get k feature maps. So one feature map could be for the blurring one, one could be the sharpening one, one could be the edge detector and so on right. There are various such filters that you could apply.

(Refer Slide Time: 15:35)

Question

- In the 1D case, we slide a one dimensional filter over a one dimensional input
- In the 2D case, we slide a two dimensional filter over a two dimensional output

a	b	c	d
e	f	g	h
i	j	k	l

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 11

Now, in the 1 D case, we slide a 1 dimensional filter over a 1 dimensional input, in the 2 D case, we slide a 2 dimensional filter on a 2 dimensional input. What would happen in the 3D case?

(Refer Slide Time: 15:45)

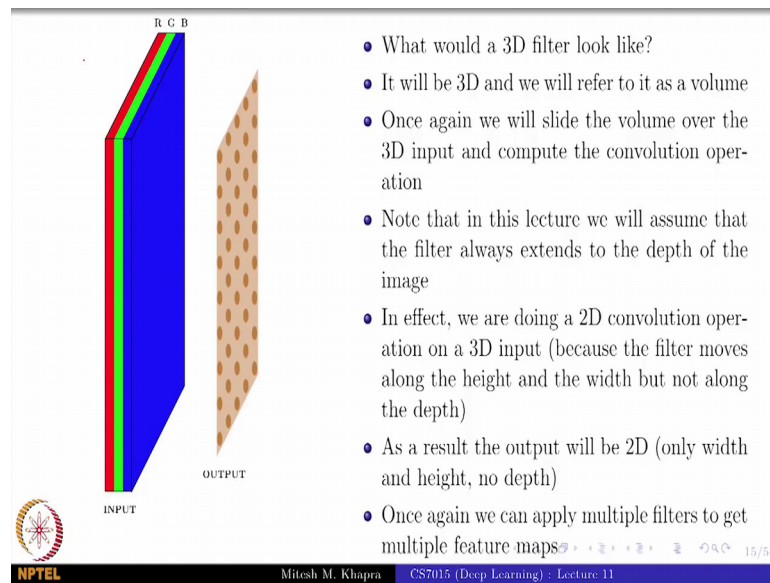
Question

- In the 1D case, we slide a one dimensional filter over a one dimensional input
- In the 2D case, we slide a two dimensional filter over a two dimensional output
- What would happen in the 3D case?

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 11

So, now, we are going to this RGB images right. So, we will have 3 cross m cross n as the input, what would happen in the 3 D case not 3 G?

(Refer Slide Time: 15:55)



- What would a 3D filter look like?
- It will be 3D and we will refer to it as a volume
- Once again we will slide the volume over the 3D input and compute the convolution operation
- Note that in this lecture we will assume that the filter always extends to the depth of the image
- In effect, we are doing a 2D convolution operation on a 3D input (because the filter moves along the height and the width but not along the depth)
- As a result the output will be 2D (only width and height, no depth)
- Once again we can apply multiple filters to get multiple feature maps

So, what would a 3 D filter look like?

Student: Box.

Look like a cuboid, a box basically and we will call it a volume, why volume? Because it has a width, it has a height and it will have a depth. So, this is what a 3 D filter would look like. I will assume that its depth is the same as the depth of your input. What is the depth of your input in this case?

Student: 3.

3. So, I will assume that the depth of the filter is the same as a depth of the input and the width and height could be 3 cross 3, 5 cross 5, 7 cross 7, anything right. So, we will get into that in more details later on. So, once again we slide this volume across the entire image ok, what is the output going to be, 2 D or 3 D?

Student: 2D.

Why? So when I was 1 D I was getting 1 D output, when I was 2 D I was getting 2 D output, 3 D again 2 D output why? Because I have assumed that, no not width.

Student: (Refer Time: 16:49).

The depth of the filter is the same as the depth of the input. So now, you just imagine this if you can suppose the filter was of depth 2 instead of 3 then, I would slide it horizontally first, vertically and then across the depth also. So, then what would be output be in that case?

Student: 3 dimensional.

3 dimensional and it would have depth of 2.

Student: 2.

Everyone gets that right, but for this lecture I am always going to assume that the depth of the filter is equal to the depth of the input always right and that is how it is for all the convolution neural networks, that we will see the depth of the input is going to be equal to the depth of the filter, the rather the depth of the filter is going to be equal to the depth of the input. So whenever, I apply a 3 D filter I am actually doing a 2 D convolution because, I am moving only along the width and the height, I am not moving along the depth. So, the output is going to be 2 D. So now, can I have multiple such filters? Yes each filter will give me a 2 D output, if I have k such filters I will have a.

Student: (Refer Time: 17:54).

k cross 2 D output right k 2 D outputs fine.