Deep Learning
Prof. Prabir Kumar Biswas
Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology, Kharagpur

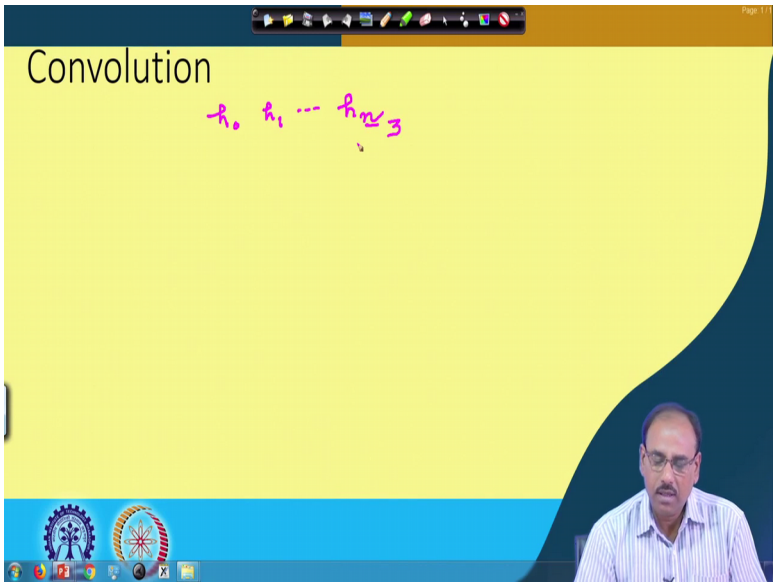
Lecture – 35
Cross Correlation

Hello, welcome to the NPTEL online certification course on Deep Learning. In our previous class, we have started discussion on convolutional neural network, and as convolutional neural network fully depends upon an operation known as convolution. In the last class, we have talked about the convolution, how to compute convolution. In today's lecture, we will talk about a very close operation which is very close to convolution known as correlation or cross correlation and in many cases because the operations are so similar mathematically that convolution and correlation or cross correlation, they are confused.

(Refer Slide Time: 01:19)

So, I will just briefly go over what we have done in the previous class. We have talked about the linear time invariant system in case of time domain signals. When it is spatial domain signal like images, the equivalent system is what is known as linear shift invariant system. And we have defined convolution in linear time invariant systems as well as linear shift invariant system.

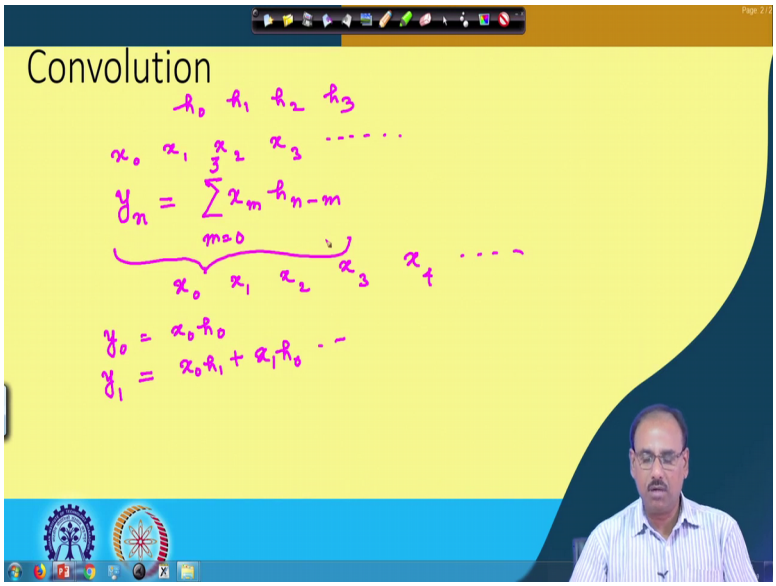
(Refer Slide Time: 01:51)



So, what we have seen in our last class that a convolution operation basically tells you that if you have a linear time invariant system, so now onwards the linear time invariant system or linear shift invariant system, we will use them interchangeably. So, when I talk about linear time invariant system, this is applicable for time domain signals like speech signal and all; but when we talk about the linear shift invariant system, it is applicable to spatial domain signal which is typically an image.

So, convolution operation is actually defined in case of a linear time invariant system or a linear shift invariant system. So, we have said that if a linear time invariant system is characterized by its impulse response, then given an input signal x to that linear time invariant system, the output of the linear time invariant system or the response of that LTI system is nothing but the convolution of the input signal x with its impulse response h. So, we have taken a discrete system where the impulse response is given by say h 0, h 1 up to say h n and so on. So, let us assume that this is the impulse response is finite. So, you have a finite impulse response system, and let us assume that the value of n is suppose 3.

(Refer Slide Time: 03:43)

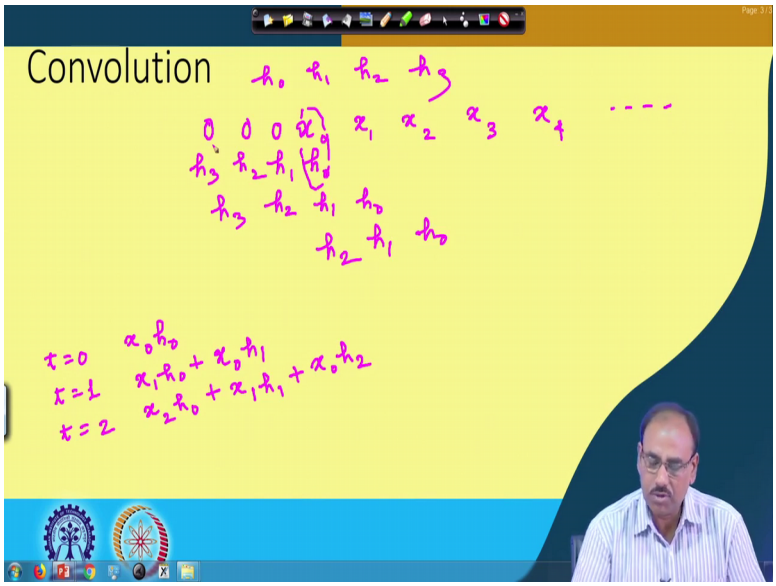


That means the impulse response is given by h 0, h 1, h 2 and h 3, this is the impulse response of the system. And suppose we have the input sequence which is given by x 0, x 1, x 2, x 3, and it continues. So, this is the input sequence. So, at time instant 0, you have input x 0; at time instant 1 you have x 1; at time instant 2, x 2 and so on.

And we have seen that given this input signal and given that output sequence, the given the impulse response, the output of the system is actually given by y n is equal to x m h n minus m take the summation over m varying from 0 to infinity. And in this case, because we are talking about the linear the finite impulse response, so m will vary from 0 to 3 as the impulse response varies from h 0 to h 3.

So, what does it mean? When I try to compute the impulse response, so given this sequence of signals x 0, x 1, x 2, x 3, x 4 and so on at time t 0, your output y 0 is given by x 0 times h 0, at time instant y t 1 the output y 1 is given by x 0 h 1 plus x 1 h 0 and so on. And that is from where we get this equation that y n is equal to x m h n minus m, where m varies from 0 to infinity or in this case m varies from 0 to 3. So, when you go for computation of this response, computation of the output or the convolution operation, the computation is something like this.

(Refer Slide Time: 06:09)



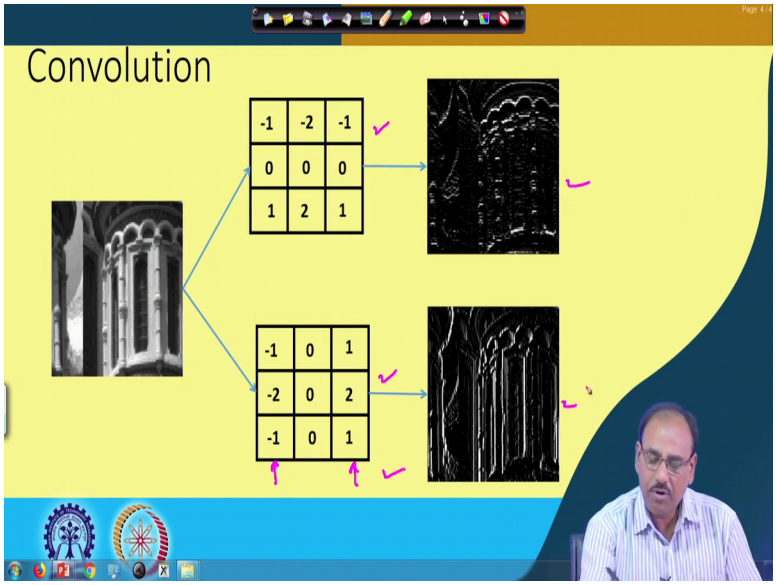
So, I assume that my input is x 0, x 1, x 2, x 3, x 4 and so on it continues this way. And when my impulse response was given as h 0, h 1, h 2 and h 3 for computation over here, I just flip it. So, what I put is at t is equal to 0, my impulse response is put like this, h 0, h 1, h 2 and h 3. So, that is what gives you that at time t equal to 0, my output is x 0 times h 0. At time t is equal to 1, what I have is this h 0 is shifted over here, h 1 is shifted over here, then I have h 2, h 3. So, at time t equal to 1, your output becomes x 1 times h 0 plus x 0 times h 1.

Similarly, at time equal to t equal to 2 h 0 shifts over here, h 1 shift over shifts over here, h 2 shifts over here. So, my output becomes x 2 times h 0 plus x 1, this is x 2 times h 0 plus x 1 times h 1 plus x 0 times h 2, and it continues like this. So, as it appears that the impulse response is flipped, and then it is shifted to location n. And at that location you compute from the impulse response and the input samples, you take their product multiply them point by point and then add them together. And because before time t equal to 0, I do not have any input symbol any input sample. So, what is assumed is that you pad a number of samples, which are all equal to 0, so that your computation is complete.

So, in this case, at t equal to 0, your computation is x 0 times h 0 plus 0 times h 1 plus 0 times h 2 plus 0 times h 3 and so on. So, this is what is known as padding. So, in case of this time domain signals, all the samples before t equal to 0, I am assuming that those

sample values are equal to 0. Similarly, when it comes to images, where the convolution operator will be a two-dimensional operator, we will also have padding where the padding will be in the form of adding extra columns, and at extra rows where all the components or all the elements in those extra columns or extra rows will be equal to 0. So, this is how you can compute the convolution.

(Refer Slide Time: 09:41)



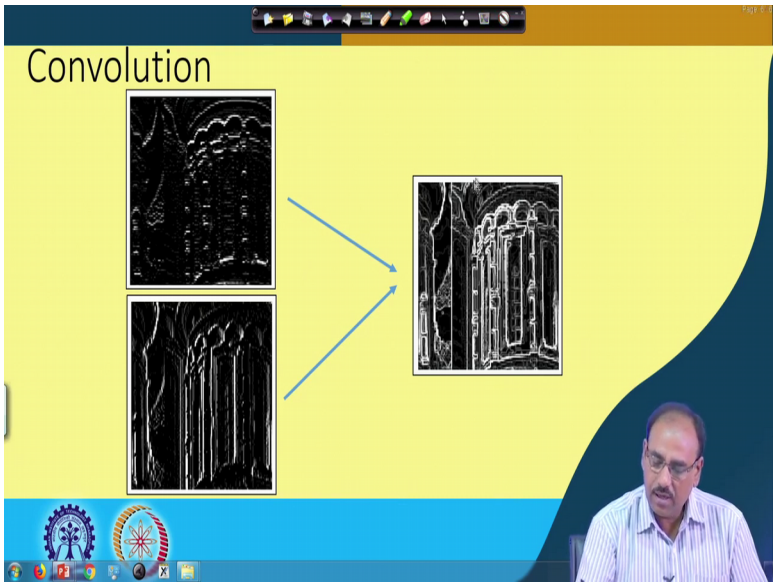
And we have seen in our previous class that given a convolution operator like minus 1, minus 2, minus 1, 0, 0, 0 and 1, 2, 1, what this operator does is it performs and weighted sum along row. And given three consecutive rows you take the take the first row, takes the third row performs weighted sums of the elements in the first row performs weighted sum of the elements. In the third row and take the difference of these two these two weighted sums or in effect that kind of operation that you are doing is differencing in the vertical direction.

Similarly, this second operator, it performs in the same operation in the orthogonal direction that is this row performs weighted sum of the first column, and then negated. This column performs weighted sum of the third column, and then finally, the inter convolution operation gives you the difference of these two weighted sums that means it is performing a differencing or differential operation in the horizontal direction.

So, as a result this operator tells you that what are the vertical edges or what are the horizontal edges which are present in the image as given over here and this operator tells

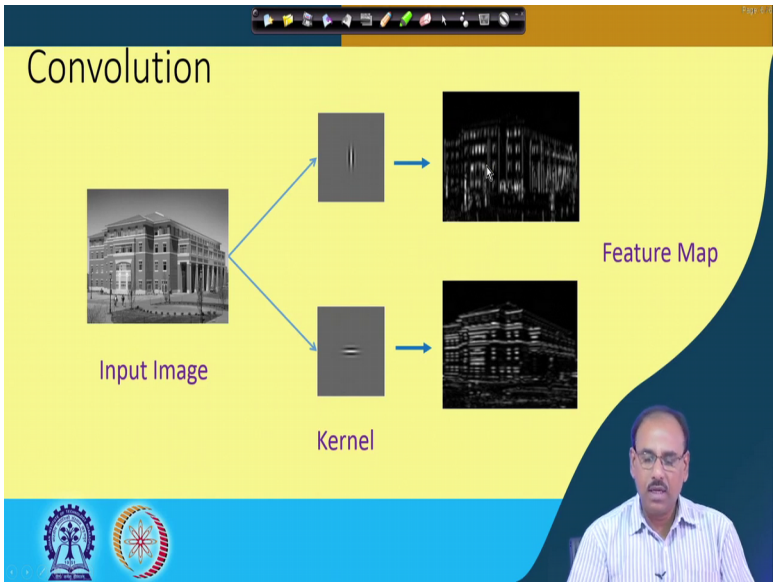
you, what are the vertical edges which are present in the image which is as given over here. So, these edges are the important features of an image, which helps us to understand or to recognize the objects present in the image. So, this is one convolution with a type of convolution operator or convolution kernel.

(Refer Slide Time: 11:41)



And if you just combine these two outputs, then what you get is this. So, here what I have is this output is just combination of these two they are merged together in some form. And you find that in this particular output or the processed image convolved image contains both the information of vertical edges as well as horizontal edges.

(Refer Slide Time: 12:11)



Similarly, if I take another kernel as given over here, so where the image on the left this is your input image. This is one convolution kernel. So, if you convolve your input image with this convolution in kernel, then this is the convolved image that you get. Here again you find that the kind of information or the features that you have are a sort of vertical edges, which are present in the image. On the other hand, if I convolve the same image with this kernel, the output, the convolved image that you get as shown over here. Again it tells you a set of horizontal edges something some information about the horizontal edges, which are present in the image. And these output images that you get these are what are known as feature maps.

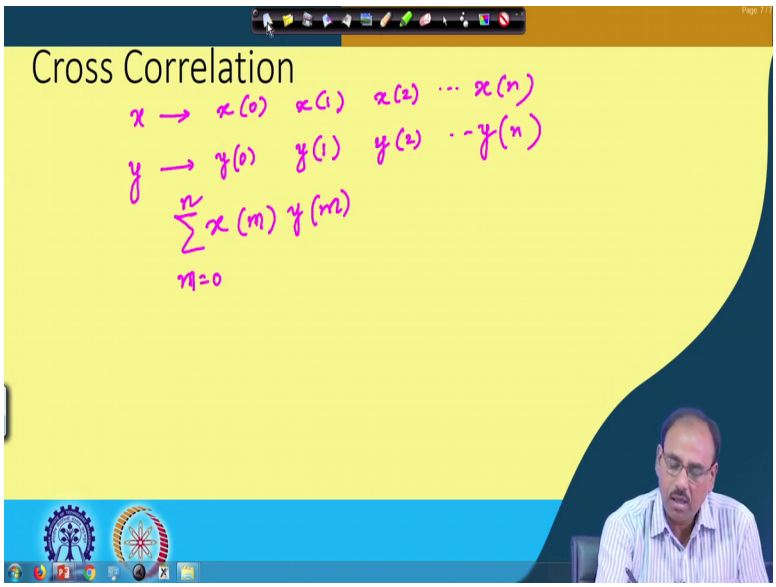
So, for your understanding of the image or to recognize what are the objects which are present in the image, these feature maps are very, very important. So, the further processing are actually done on the feature maps in deeper layers of the neural network that we will come to later on.

So, we have seen that in case of an one-dimensional image, you flip the convolution kernel, and then shift it along the samples input image samples, and then you perform the convolution. Similarly in case of images where my convolution kernel is a two-dimensional convolution kernel again, I have to flip it before I perform the convolution operation. And now the flipping has to be both around vertical axis and around horizontal axis.

So, once the kernels are flipped around vertical and horizontal axis, after that the kernel is used for convolution operation. But one thing you should be clear about that this convolution operation is not the operation between two different signals, but the convolution operation actually gives you the output of a linear time invariant system or linear space invariant system, where the convolution kernel is actually the impulse response of the system.

So, given this impulse response whenever you have a an input signal, whether it is a time domain signal or one-dimensional signal or it is an image which is a two-dimensional signal, the output is actually which is the convolution of your given signal with the convolution kernel. The output is actually the response of the system characterized by the impulse response to the input signal that we are giving. So, this is what the convolution operation is. Now as I said that a very close operation, which is very close to convolution which is known as correlation.

(Refer Slide Time: 15:21)

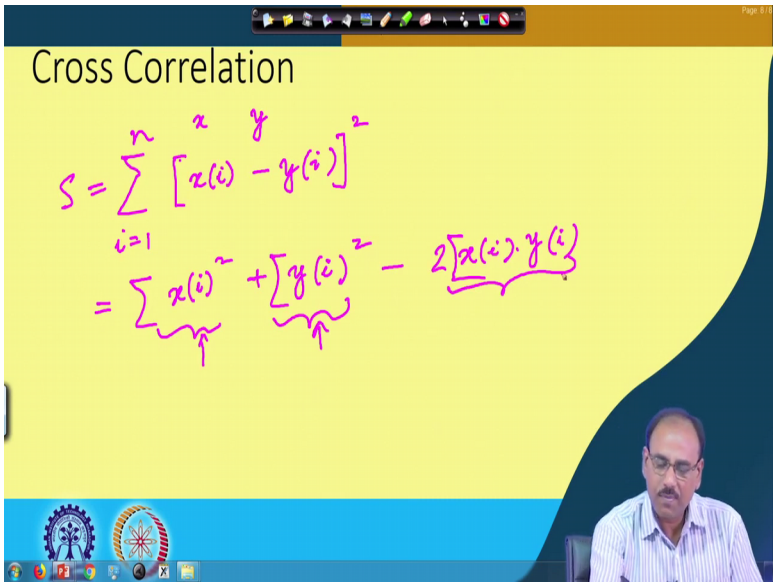


So, when you find the correlation between two different signals, this is what is known as cross correlation. So, unlike in case of convolution, where the convolution output is the response of the system to a given input, in case of correlation the cross correlation actually tells you what is the similarity between two given signals. So, here cross correlation is the actual operation between two given signals, whereas the convolution is not.

So, the correlation operation is obtained like this. Suppose, I have one signal say x and the other signal is y ok. Again initially let us talk about the discrete signal. So, x is given by x 0, x 1, x 2 up to say x n if I have n number of samples. Similarly, for y, the samples are y 0, y 1, y 2 up to y n, where y also has n number of samples. Then the cross correlation between x and y is given by x n, y n, take the summation over n varying from 0 to sorry let me write it as x m, y m. So, where m will vary to will vary from 0 to n. So, this is what is the cross correlation between the two signals x and y. And this cross correlation is nothing but what is the degree of similarity between the two signals x and y.

So, how do we say that it is this cross correlation actually gives you the degree of similarity? You remember so far whatever we are discussing is in the discrete case obviously, this can be generalized to continuous case, where the sampling frequency can be assumed to be infinite right that is inter sample gap is infinitesimally small, then this discrete system actually leads to a continuous system. We will come to that a bit later.

(Refer Slide Time: 18:05)



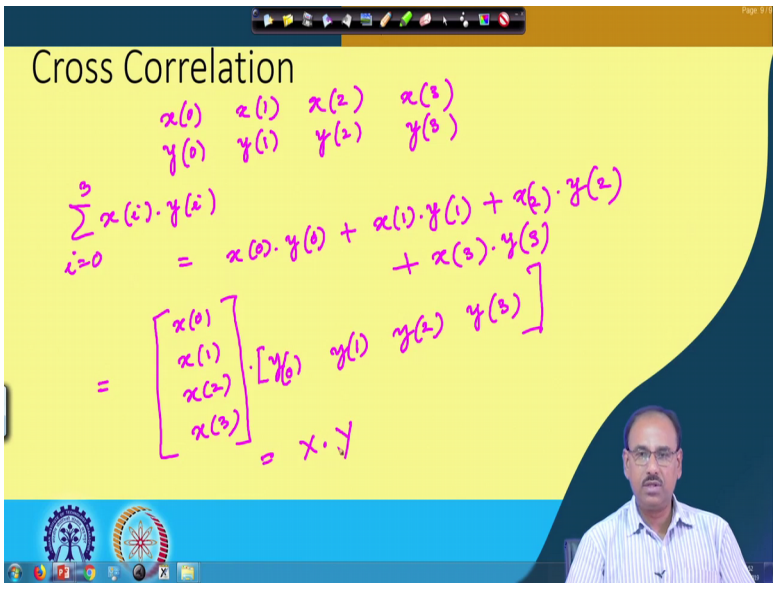
So, suppose we have this two signals x and y, and I want to find out that what is the difference between these two signals x and y. And to compute the difference, what we usually do is we take the sum of squared error that means I take x i minus y i, take the square of this, and the sum of this for I is equal to 1 to n. So, this is the sum of squared error between the two signals x and y. And obviously, if the signals are widely different,

then the sum of squared error will be high; if the signal is very close x and y are very similar, then sum of squared error will be low.

So, here you find that if I expand this, this simply becomes x i square plus y i square minus 2 of x i into y i, take the summation of this, summation of this and summation of this. So, here you find that given x and y that is your given a set of samples from x, you are given a set of samples from y. So, given x and y, sum of x square and sum of y square is fixed. And if the two signals x and y are very similar, then sum of squared has to be low. So, if sum of squared error has to be low and given that x i square and y i square they are to be fixed, I must have for smaller values of sum of squared error S. So, so, let me put it as S. So, for smaller values of S, I must have x i into y i sum of that to be very large.

So, that clearly indicates that if x and y, they are similar, then sum of x i into y i will be large. Whereas, if the signals are widely different, then obviously S has to be very high if s is high that is sum of square error is high, then x i into y i summation of this over i that has to be low. And this x i into y i sum of this is nothing but the cross correlation between the two signals x and, x and y. So, this cross correlation actually tells you what is the degree of similarity between the two signals x and y. Now, what is this x i into y i sum of this.

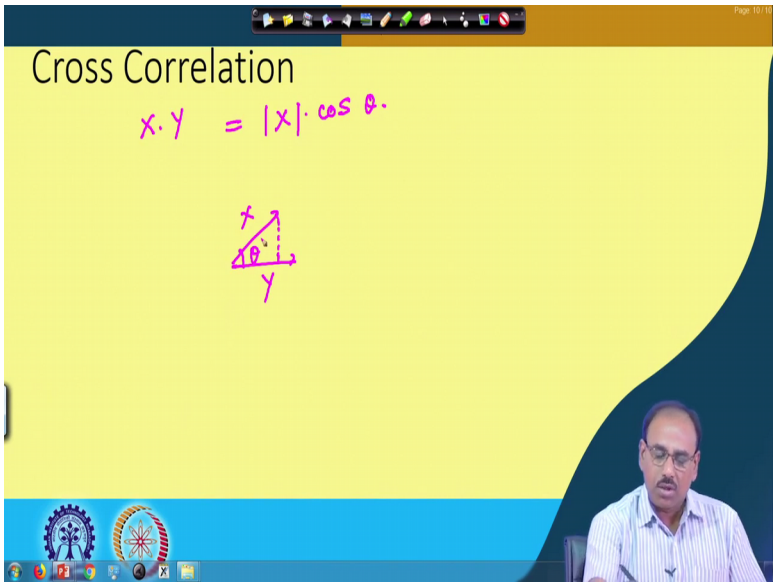
(Refer Slide Time: 21:23)



If you think that x i have signal say x 0, x 1, x 2, x 3, so value of n is equal to 3, and the other signal is y 0, y 1, y 2 and y 3. Then x i into y i summation of this, i varying from 0 to 3 is nothing but x 0 into y 0 plus x 1 into y 1 plus x 2 into y 2 plus x three into y 3, which is nothing but I can put it in this form this is x 0, x 1, x 2, x 3. If I put it in the form of a vector representation times y 0, y 1, y 2, y 3. So, if the state of input samples of X is considered to be a vector, and similarly set of the samples of Y is considered to be another vector say Y, then this cross correlation is nothing but the dot product of X and Y.

And we know that if the vectors X and Y are similar, then the dot product between the two vectors X and Y will be high; if the vectors are dissimilar, then the dot product will be quite low and that is what the cross correlation tells you. And, in fact, if one of the two vectors X or Y is an unit vector, then this dot product straightaway tells you what is the degree of similarity. So, suppose let me take just a two-dimensional case, that X is a two-dimensional vector, Y is also a two-dimensional vector, and say Y is represented by an unit vector.

(Refer Slide Time: 23:45)

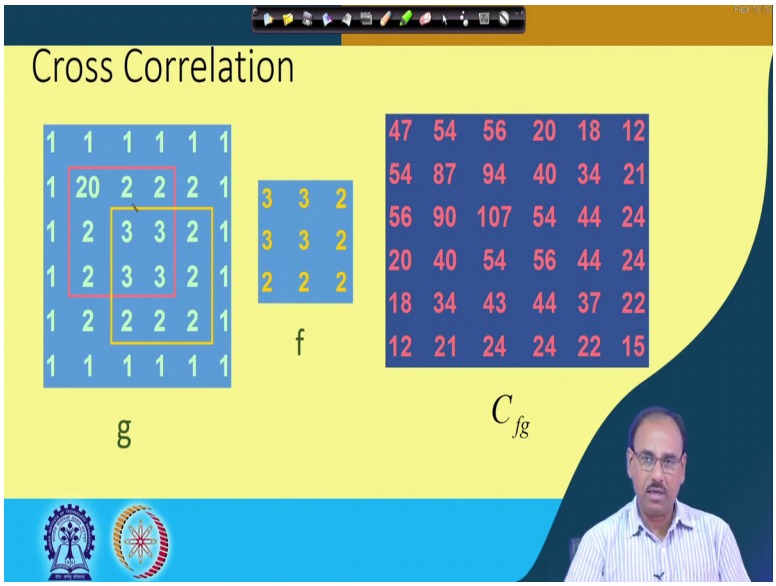


So, if I take X dot Y; where Y is an unit vector is nothing but in two-dimensional case this will be mod of X into cosine theta, where theta is the angle between the two vectors X and y. And if X and Y they coincide that means, they are in the same direction, then cosine theta is equal to 1, so the dot product is simply mod, mod of x. As theta increases from 0, then cosine theta goes on reducing it reduces from 1. So, the value of the dot

product also goes on reducing. So, it is simply like this that I have vector Y in this direction which is an unit vector, and I have X over here, so this X dot Y is nothing but projection of X onto the vector Y, and this is the angle theta between the two vectors.

So, as theta goes on increasing, the projection the value of the projection of the length of the projection goes on reducing. And when theta is equal to 90 degree, the projection is 0 when the vectors are orthogonal to each other. And this dot product or the projection simply tells you what is the degree of similarity. And that will be make maximum when theta is equal to 0 that means X is in the direction of Y, or X is a multiple of vector Y. So, given this now the question comes that how do we compute the cross correlation?

(Refer Slide Time: 25:27)



So, let us give some example of this cross correlation so over here. So, as cross correlation tells you the degree, degree of similarity between two signals, the cross correlation can also be used if one of the signal, I take as a template, and another signal is a bigger signal the cross correlation can be used to find out in a bigger image, where the smaller image given by template exists ok.

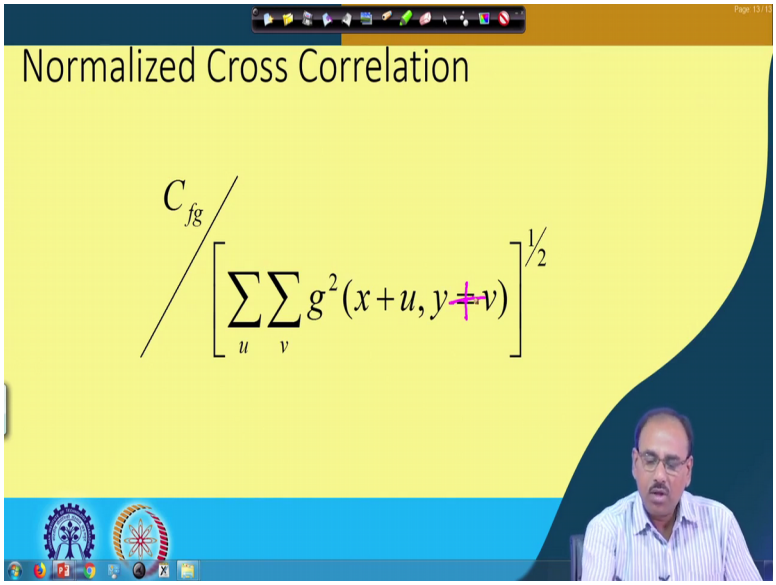
So, given that I am taking an example over here, this is say a bigger image. And this is my template I want to find out that in this bigger image, where this template exists. So, for that I put this template over this image and shift it in every position for every position of the template I compute cross correlation. And I compute cost correlation value for all different positions of the template and in a particular position where the value will be

maximum that is the location, where I can say that the template exists within the given image.

So, given this and given these two images f and g as given over here the cross correlation matrix that I get is as given by the C f g. And if you look at C f g, this is the one which is the maximum value. So, this is where I should expect that my template exists within the given image. But if you look at this actually it is not so, this is which corresponds to this maximum value 1, 0, 7, whereas my template actually exists over here which is not given by this cross correlation.

So, what is the fallacy in my computation? The fallacy is when I am computing the cross correlation at different locations, my template f is fixed, but template g is not. So, the cross correlation value that you compute is actually biased by template g by the part of the image, which is g. So, what I have to do is as I said that if one of the vectors is an unit vector, then I actually get the similarity index. So, here I have to normalize the part of the image given as g with respect to g itself, so that it is equivalent to a unit vector. So, it is a normalized image.

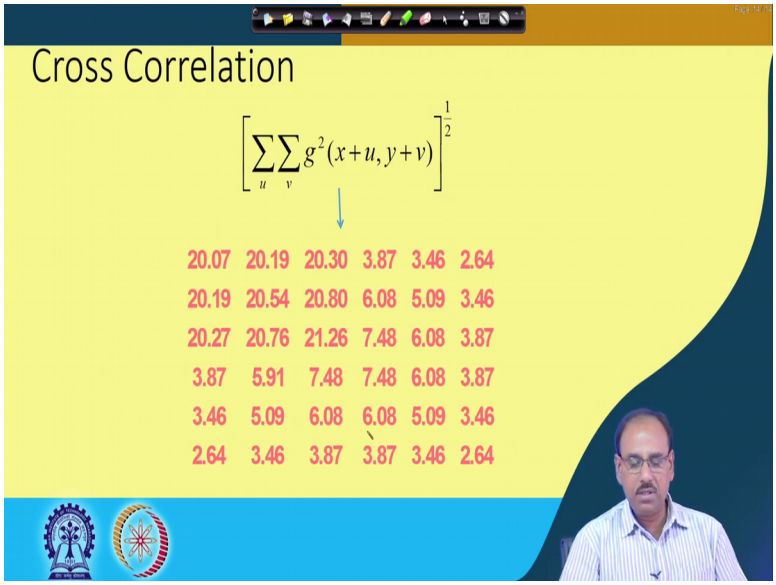
(Refer Slide Time: 27:55)



So, what I have to do is I have to normalize the cross correlation by the value of g squared x plus u y plus v this should be y plus sorry, it is not equal this will be plus. So, x plus u y plus v g square of that take summation take square root of this so, with this I

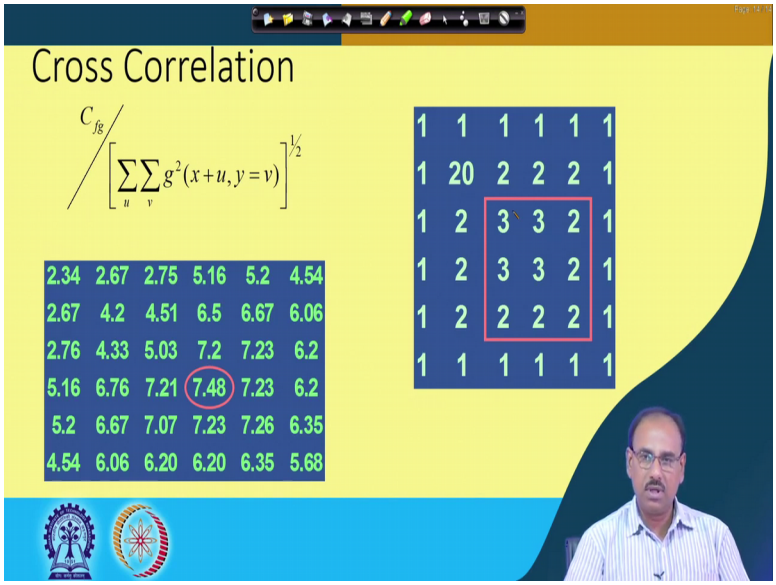
have to normalize my computed cross correlation. So, that it becomes independent of the value of g, which is variable. So, once I do that, let us see what we get.

(Refer Slide Time: 28:35)



So, for the given image if I compute this g square x plus u y plus v take the summation over u and v and then square root of this, then this is the matrix that you get.

(Refer Slide Time: 28:49)



And now if I normalize by computed cross correlation with this normalization factor, what I get is normalized cross correlation matrix. And in the normalized cross correlation matrix, you find that this is the location where the normalized cross correlation value is

maximum. And corresponding to this, this is the location in the bigger image or it shows that I have my template present in the given image and this is the perfect one.

So, what we have seen is in case of convolution, I get the response of the system to a given input; in case of cross correlation, I get the similarity between two different images or two different signals. In many cases these two are confused the reason is computationally the way you compute cross correlation and the way you compute the convolution, they are similar. The only difference is in case of convolution, you have to flip your kernel around vertical and horizontal dimensions before you compute something very, very similar to cross correlation. So, in minimal of the books, you will find that what the present as convolution is nothing but cross relation, and as I said that computationally they are not different.

And I also said that both convolution and cross correlation, we have discussed with respect to discrete systems. This can very well be generalized to continuous systems, where the summation will be replaced by the integration, and obviously, I will have an integral term like dt, or dx, dy ok.

So, we will stop here today. So, I hope that you know, what is the difference between convolution and cross correlation and with this starting from our next class, we will start discussion on convolutional neural network.

Thank you.