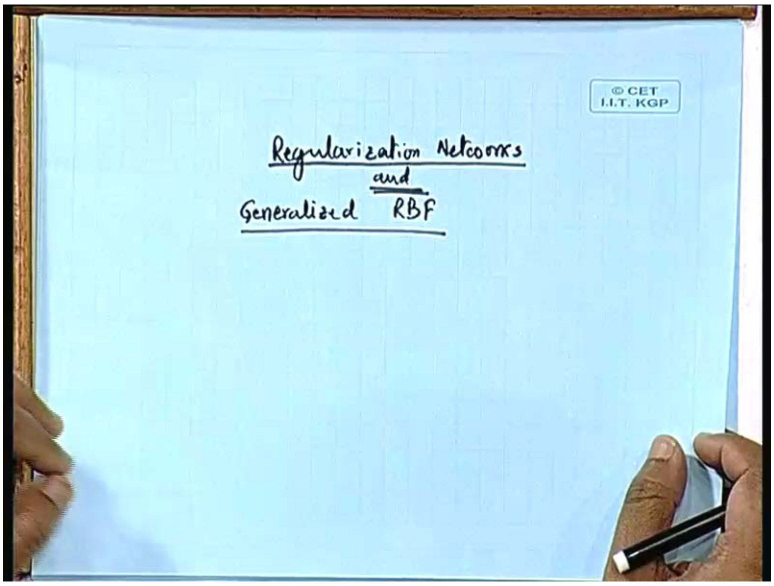
Neural Networks and Applications
Prof. S. Sengupta
Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology, Kharagpur
Lecture – 29

Regularization Networks and Generalized RBF

Today's lecture, is going to be on the Regularization Network.

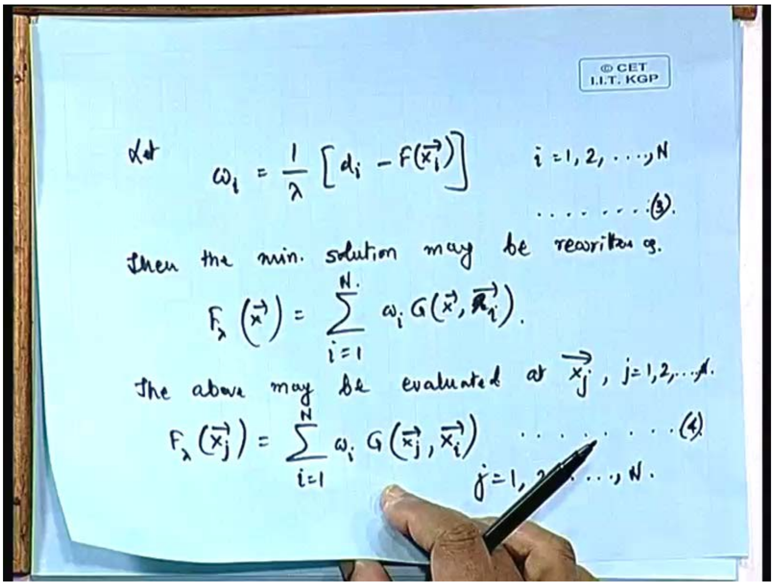
(Refer Slide Time: 01:00)



Regularization networks something, which we intended to discuss in the last class itself and we have already developed the necessary theories pertaining to it, but we will be showing that how the regularization networks really looks like. And then, we will be going over to the next topic that is generalized radial basis function. Again, I think, we have been spending quite a few lectures on the radial basis function, I think last 5 or 6 lectures have been devoted to the RBF and we have to cover some more of the topics.

Before the end of this semester course load, that is 40 lecture courses, so we have to draw our conclusions, etcetera, quite soon about the RBF and proceed to next chapter. Anyway, so today, I think we are going to exclusive devote this topic that we have, as our plan for today. Now, you remember that in the last lecture, when we were discussing about the regularization, the final solution that we had presented was of this nature, that we were of obtaining, that concentrate on the equation 4; that we had got where, the function F lambda of x j, x j being the input vector.

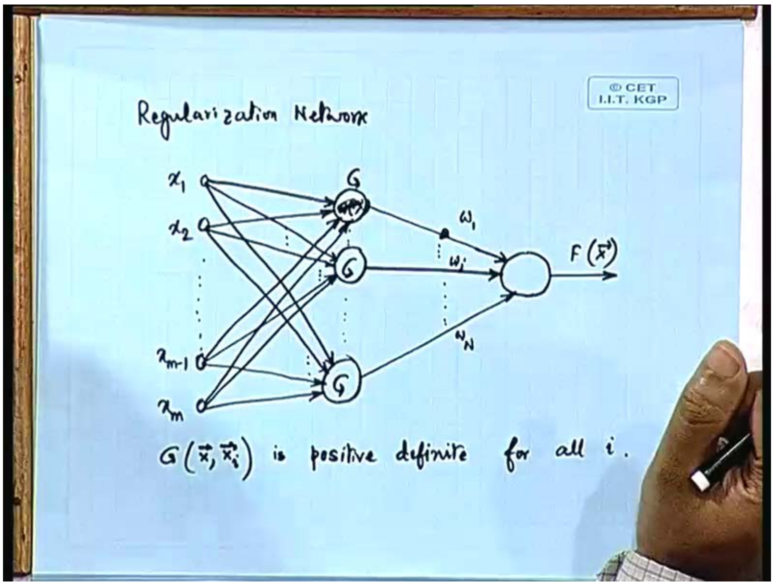
(Refer Slide Time: 02:32)



That is actually expressed as a summation of w i times G of x j comma x i, where G happens to be green function, center around x i. And what we have to do is that, we are finding the response of the function at the point x j, so it is G of x j comma x i with x i as the center. And with the weights w i, which is nothing but the difference between the d i and the F x i, the desired and the actual output and it is divided by 1 upon lambda.

Well, lambda is regularization parameter, so this is the found, that we had got summation of i is equal to 1, 2 to N. And from this, we can construct the regularization network, where you can well understand that the inputs will be the x i's, the patterns will be the input. And as the output we will be getting this F lambda of x j, so if we see that the structure of the regularization network, it should look something like this.

(Refer Slide Time: 04:07)



Let say that, we have got the inputs, where these are x 1, x 2, so these are the individual elements of the input vectors. So, when I write as small x 1, small x 2, etcetera, they are nothing but the individual inputs, this is x m minus 1, which is the last part 1. And the last is x m, where we are assuming m to be the dimensionality of the input and we are going to have the green's functions realized here.

So, here we could have some green's function realizations and can you tell me that, how many green's function, we have to draw out here. How many green's function equal to N, where n is the number of patterns that we are feeding to the system. So, each of the pattern is going to be m dimension of vector and we are having N such patterns and the equation, which we had return, you see that here, the summation is for i is equal to 1 to N and x j comma x i.

So, what we have to do is to add of the responses of N different weighted responses of these green's function that is what we are adding up. So, what we have to do is to have here N number of such green's function computational blocks, where the centers will be like x 1, x 2, etcetera up to x m. So, this will be the green's function G, were we will be having green's function of sum inputs, but center around x 1. Like that, this will be let say center around some x j and this is center around some x N.

And what we have to do is that for each of this green's function computational unit, we will be having all the m inputs, all the m elements of the input, we have to be pay to these G function, because the G will be compute over that vector. Because, x 1 to x m will compose the input vector and G will be computed on that vector, center around another pattern vector, which is there.

So, what we have to do is do draw the connections like this, here we draw the dotted lines indicating that, there will be all together m such inputs. And very similarly, all these inputs will be connected to the other green's function networks as well. Similarly, the last 1 also, we should receive x 1, x 2 up to x m minus 1, lastly x m. So, these will be the responses of the green's functions computational unit.

And what do we have to do, they have to simply weight it, weight the individual weights, so this have to be multiplied by w 1, in fact, we do not draw the arrows out here. So, this will be another unit, which will be linear, linear computational unit will be kept over here. Where, this output, that we are having, this is the first green's function output, this will be multiplied by w 1.

The second green's function like was will be multiplied by w 2 and if these 1, we take as the jth green's functions computational unit, then this has to be multiplied by w j. And if this is the last green's function computational unit, that is G center around x N and then this response has to be multiplied by w n. And all these responses will be added of in this linear input, in this linear unit and linear computational unit and then, we will be having F of x factor.

And what is the x factor, x factor is nothing but the vector representation of the inputs that we are feeding. So, this is the structure of the regularization network, now this has got 1 assumption of course that the G of x comma x i. I do not have space to write down this, so instead of making the G's look clumsy, I simply write G, although I try to write here. So, let me write G within these boxes and then, actual computation will be G of x comma x i.

And the essential condition for this is that, this has to be positive definite for all i, so it is quite simple in structure, in fact this is the structure of the regularization network. It is nothing but the simple network representation of the equation that we had got, the equation number 4 that we have got. As expressing the F of x j as summation, it is just a network representation of this.

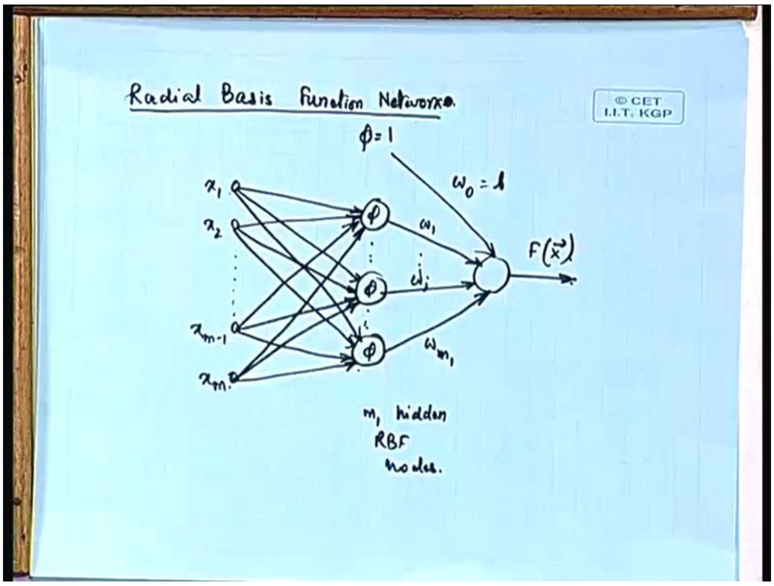
Now, this looks that is similar to what radial basis function network looks like, now we have discussed about the radial basis function, without of course really drawing the actual network structure. Now, here you can see that the number hidden nodes, that we are having in this network is, that there are N number of hidden nodes. Where, N is equal to the number of patterns that we are feeding to this.

Whereas, in the radial basis function network, we are not having n hidden nodes, we are not putting that restriction, that it has to be N. We can have a number of hidden nodes, which is equal to m 1. Now, we discussed the typical m 1 has to be of a dimensionality had, then that of m, that is the input dimensionality, because normally, we are mapping from a lower dimensional inputs space to relatively higher dimensional hidden spaces.

So, instead of N, if we take the number of hidden nodes to be m 1; that the total number of hidden nodes, then the radial basis function network would look something very similar to this, in fact there the G is will be replaced by phi's. In fact, I discussed last time also, that the difference between the phi's and the G's are, that in the phi that is the radial basis function, we are simply doing the interpolation where, we are putting restriction. That the surface that we are going to interpolate is going to pass through the given set of points.

Whereas, in the case of regularization, we are not putting that construct, what we are instead sayings that the, surface that we are going to reconstruct has to minimize functional, that takes into count of both. That is to say the actual cost function as also the variation, that is to say smooth differential; that is what we showed. So, now, very similarly radial basis function network would have look like this.

(Refer Slide Time: 12:38)



So, there also we would have got x 1, x 2 up to here x m minus 1 and the last is x m and then, we would have got here the hidden neurons, the hidden nodes where, there will be m 1, such hidden RBF nodes, so this all well computed the radial basis function. So, we can write here phi, indicating that, all phi has computing the radial basis function. And very similar to what we had done, for the regularization network, the radial basis function network will contained this.

In fact, this realization directly follows from a very similar series that we had obtained earlier. I think few class back, we had obtained a very similar series of F x equal to i is equal to 1, 2 m 1, w i F, the F matrix that we had got. So, it is simply that that realization, so x 1 to x m going over to every phi's and then, we are going to have the output. And in fact, here the output neuron will not only add up this, but also we are going to have a bias.

So, here if we take phi to be equal to 1, then we will be having here w 0, which is equal to the bias and then we will be having here w 1. And if this is the jth radial basis function this is w j and this is the last basis function, last RBF computational units. So, this has to be w, m 1, this has to be w, m 1. And this is going realize the F of x, so this one is actually call the radial basis function network.

So, these looks very similar to got the regularization network is, only thing is that here instead of m 1, we are considering N. Another question, that because, if we have to obtain the solution of the regularization equation. Then, the optimal solution that we had obtained definitely had got N number of such terms where, N is the number of patterns that we had pay to the system.

And in fact in the matrix representation also, you could very easily see that we had got the matrix that involve the G matrix. Now, G matrix is essentially and n by n matrix and then, we had to obtained and inversion of that the G plus lambda I, we have to obtained the inverse matrix to that. And that 1 was having actually, if when N is very large, then this requires actually N cube number of computation, for the computation of the inwards to that matrix.

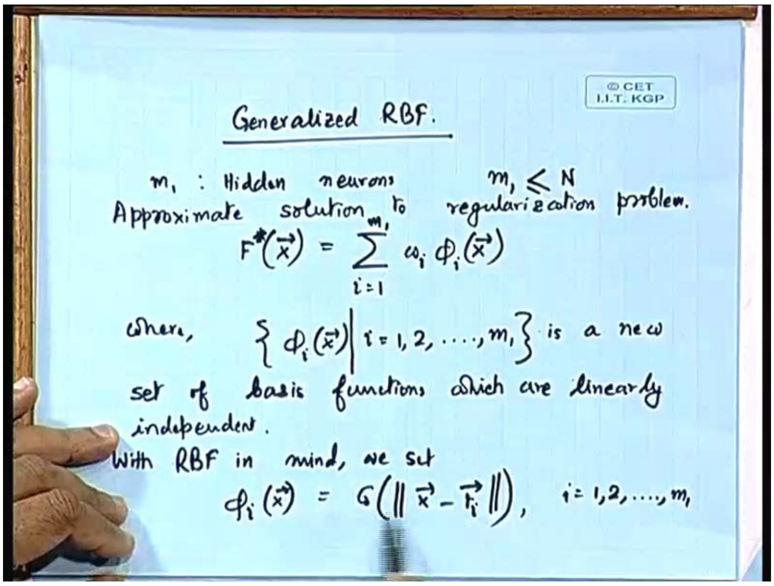
So, if N, the number is very large, if the number a pattern is extremely large, in that case, the computation of the regularization involving the green's function, requires N norms amount of computation. Because, it will polynomially increase with N and typically, N is going to be large, because if our given set points is large. And the essential requirement there is, that we have to designed, that with every such input pattern, we have to design a green's function that center around that pattern.

And in that case, only we can obtain an optimal solution, now the question is that, if we impose a practical implementational restriction to it is, especially in the case, where this n is going to be very, very large. In that case, while designing the network, instead of designing with N number of hidden neurons, we arrive at a figure, let say m 1, which is typically, much less as compare to that of N.

And if we can obtain network with m 1 number of hidden nodes using this function only, if we obtained m 1 number of such hidden nodes. And if we can solve the regularization network, that does not involve N number of terms. But, rather m 1 number term, but m 1 is the number up to which, we go. Instead of taking all the N green's function, we take and m 1 subset of it, we take m 1 number green's function and we try to sallow the problem.

Using those m 1 as the centers, in that case, we are not obtaining an exact solution; we are not obtaining optimal solution to it. But, definitely some sub optimal solution that is what we are getting, which could be used for the practical implementation of regularization. So, we will go into the theory of this, which is known as the generalized radial basis function.

(Refer Slide Time: 19:11)



So, we are now going to discussed about the next topic of day, which is generalized RBF or the generalized radial basis function. And the essence is that, there we avoid having N number of such hidden neurons. Instead, we designed it with m 1 number of hidden neurons. Where, m 1 is less, mathematically we should say that m 1 should be less than or equal to N. When, m 1 is equal to N, then it leads to the optimal solution, which we had of already obtained.

So, now you remember, that we had got a solution of this from, may had to obtained a solution of F x equal to summation of w i phi i x. And where we have to add up i is equal to 1 to in fact, if it is with N number of patterns, we are going up to N, but here we are going up to m 1. And in fact, as a solution to the regularization equation, if we assume function of this nature, where we are not taking N terms, mind you, we are taking m 1 number terms.

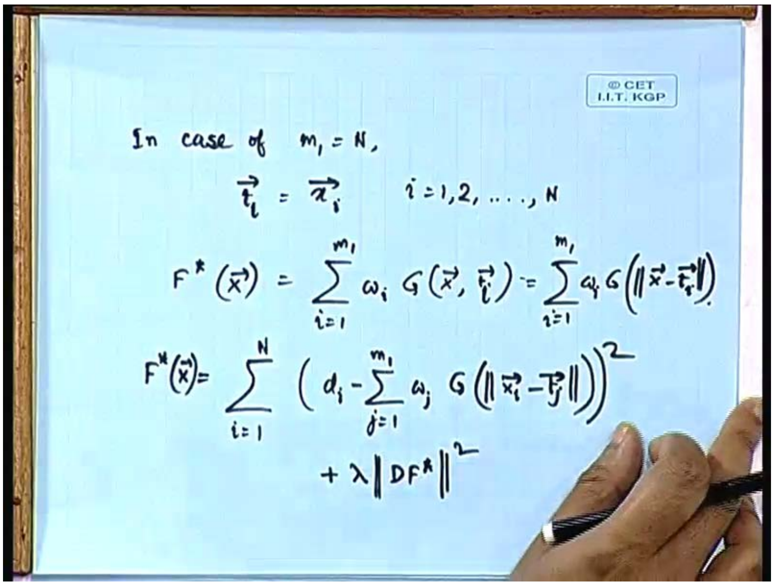
Taking m 1 number of terms obviously means that, it is not be optimal solution; it is definitely some kind of approximated solution. And as new symbol to the approximated solution, we writhed down as F star of x, so this is the approximate solution. So, this is the approximate solution to the regularization problem. Now, in this case, we set phi i that we are considering, that is a new set of basic functions.

Now, mind you, we are not writing it has G, we are writing it has phi i's, why, because with G as basis function, we required N number of terms. With G, we would have required N sub G's, but here instead of using N such G's, we are using m 1 number of sum basis functions. This may not be the radial basis functions, which we had used for interpolation purpose strictly.

But, here definition wise, we can say that, this phi i x, the set of such phi i x, with i is equal to 1, 2 up to m 1. This is new set of basis functions which are linearly independent. Now, with the radial basis function in mind, with RBF in mind, we can set the phi i x to be equal to G, the green's function of we can say norm of x minus t i. In fact, we can write either G as x comma t i or G of norm x minus t i, whatever we represent, it does not, really matter.

Now, here i is equal to 1, 2 up to m 1 not N, so these are the set of center, so what are the centers of the basis functions, all this t i at the centers. Now, this particular representation, actually guarantees that in the case of m 1 equal to N.

(Refer Slide Time: 24:22)



In the case of m 1 equal to N, we will be having t i equal to x i per i is equal to 1, 2 up to N. Whereas, when we are having a number m 1, that is less than N, when we are obtaining m 1 less than or equal to N. Then, centers here remain as t i, they are not exactly the x i, we have somehow designed m 1, such different centers, which may not be exactly matching with the patterns that we are feeding. There are m 1 such centers, indexed has t 1, t 2 etcetera up to t m 1.

Only in the special case of m 1 equal to N, t i is become equal to x I's and which case, all the N number of patterns that we are having, they are individually the centers of the green's function. Now, what we have to design as this function is that, in case of m 1 equal to N, the correct solution should be consistently recovered. In fact, that is very much possible.

Because, what we will be having, in that case is that phi i x, in such case will be equal to G of x comma x i and G of x comma x i, if we substitute over here. And if m 1 is equal to N, then we are obtaining the regularization solution that we already obtain. So, this representation is consistent with means as special case taking m 1 equal to N. This F star x will become equal to actual F x of F lambda x. But, otherwise we are taking the approximated representation.

So, in this representation, we can write down F star of x equal to summation i is equal to 1 to m 1, w i G x comma t 1, which is equal to t i sorry, x comma t i and this can also be represented as i is equal to 1 to m 1, w i G of norm of x minus t i, whichever we are representation, one chooses. Now, this could be represented, the expansion of this could be actually leading to i is equal to.

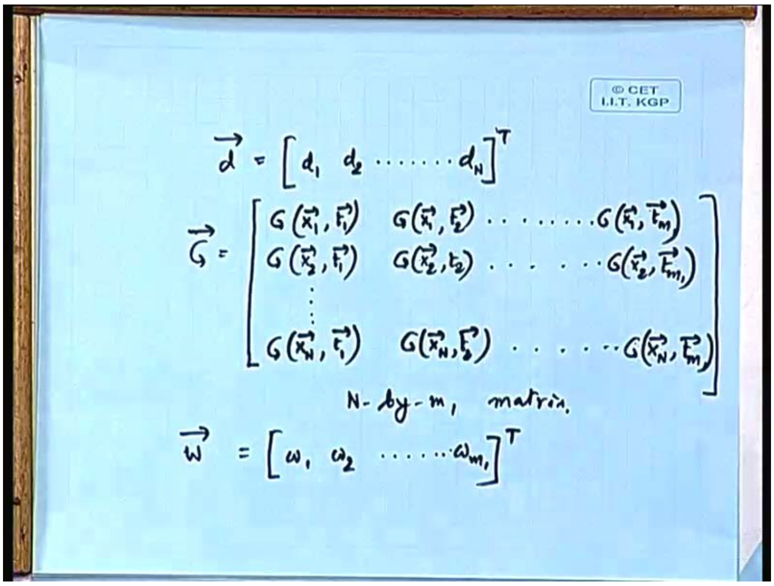
Now, the approximating function F star of x, if we take F star of x, mind you, has got two terms. The basic equations as got two terms, one is the standard error term and the next one is the regularization term, regularization term means d of F star in this case. Whereas, the standard error term is going to be, if we take the d i as the desired input, in that case, it will be what, d i minus this one, this one is going to be the actual outputs.

So, d i minus, if I take i to be the index over here, then while summing it up, we should use a different index, so j is equal to 1 to m 1 of w j G of x i minus t j. This term and we have to square this thing up and this has to be, this is the error for walk, this is the error for the ith pattern. So, we have to added of for i is equal to 1 to N, not m 1, because N is the number of patterns, that we are feeding to this system.

Whereas, m 1 is the number of hidden nodes that we had chosen, the number of green's function with which we are going to approximate the regularization solution. So, here the index is m 1, whereas outside here the summation is up to N. So, please note this, so this is the standard error term. And then, we have the regularization term as lambda times norm of D F star, this as the squared norm of this.

Now, we look at the first term of this expansion, the standard errors term, this can be easily represented in a better form using the matrix. And how do we do it, very similar to what we did earlier; that means to say that defining a d vector, where d vector will contain all this that is d 1, d 2 up to d n, the desired outputs.

(Refer Slide Time: 30:09)



They will be composing the d vector and then, here we are composing a G matrix, now G matrix you tell me, that is this is the summation representation. We are obtaining here G x i comma t j. Now, when we come to t j that is the centers, they are m 1 number of such centers, but coming to this x i that is set of first argument of this. Coming to this x I, how many x I's are there, there are N number of x I's, but there are m 1 number of t j's.

So, how many elements are there in that G matrix, that we are going to compose, m, so here, we are going to have N by m as the matrix composition. Where, N is the number of patterns that is x 1 to x N, they will compose the rows of the matrix and t 1 to t m 1's, they will be along the column of the matrix. So, we define a matrix G, which is not a square matrix any more, last time the G matrix was an N by N square matrix.

Whereas, in this case, it is going to be N by m 1, known square matrix, so here, this will be G, the first element will be G of x 1 comma t 1. The second element will be G of x 1 comma t 2; like that the last column will be G of x 1 comma t m 1. And now, coming to the second row, it will be G x 2 comma t 1, G x 2 comma t 2, the last will be G x 2 comma t m 1.

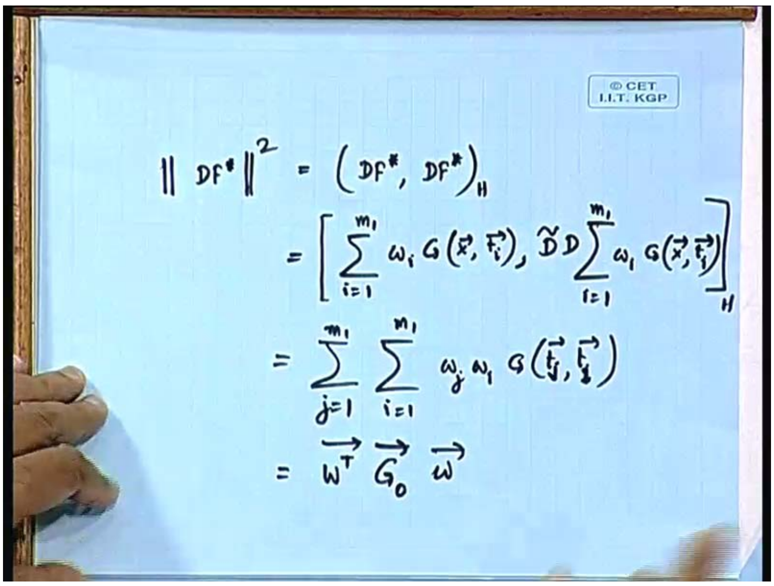
Like that, we go on the last row will be G of x N comma t 1, G of x N comma t 2; the last will be G of x N comma t m 1. So, this composes our N by N matrix, so this is N by m 1 matrix for G, not a square matrix any more. So, this is the green's function matrix and then, w which is equal to here, how many elements will be there with w vector, m 1 number of elements will be there.

So, it will be w 1, w 2 up to w m 1 determining the dimension of the w is very easy, because as many hidden neurons has you have in the system. You have to add of the responses, waited responses of all these hidden neurons So, there are m 1 number of hidden neurons, so necessarily there will be m 1 number of synaptic weights, which will be associated with it so. This will be w 1 to w m 1 and since, it is to be represented in form like this and we have to take the transports.

Now, actually one property that we are getting here is, that last time we had obtained G matrix as a symmetric matrix, whereas in this case, G is no longer a symmetric matrix, because it is not square matrix any more. And last time, we had N element w vector, here we are having m 1 element w vector, so these are the two thinks that we are observing.

Now, taking care of the second term, that is to say the regularization term D of F star, this can be represented very conveniently in the form of the inner products representation in the Hilbert's space, which we had already done last time. So, writing in this form and then, doing the simplifications, we should obtained the expressive for D F star like this.

(Refer Slide Time: 34:39)



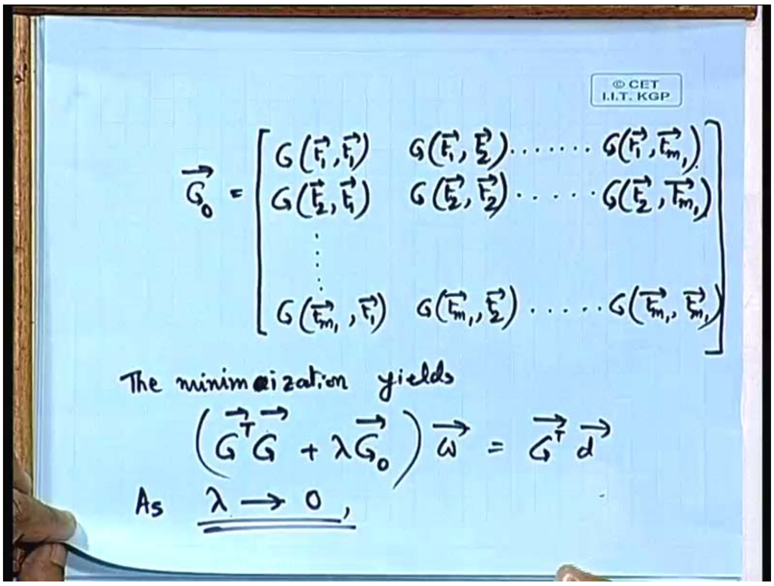
So, this is the deferential operated on the F star; that is the approximating function D F star, F star. So, taking the deferential of that and we have to computed D F star square, the square of that. So, in the inner product representation should be D F star comma, D F star, the inner product with itself in the Hilbert's space. And this could be represented now by making use of the adjoint property.

By making use of the adjoint property, we can write it as this form of inner product, i is equal to 1 to m 1, this one can be return as follows i is equal to 1 to m 1, w i G x comma t i. In fact, this from as you can see is the representation of F star, directly because F stars got, it term like this. And then, we can write it down, making use of the adjoint, we can write it down as D star D of this, i is equal to 1 to m 1, w i G x comma t i. And we take a taken inner product of this and this in the Hilbert's space.

This on simplification leads to this sort of a summation j is equal to 1 to m 1, summation i is equal to 1 to m 1, w j, w i, G of t j comma t i, sorry t j comma t i. So, this is the representation that we are getting by simplification and in fact, so this is t j comma t i, mind you. And this in the form of matrix, matrix vector one can represent it as W transpose, as the vector W transpose is nothing but the transpose of the W vector, that we have already defined.

And then, there will be a term corresponding to this G and mind you here this is a double summation that involves m 1 in the inner loop, m 1 in the outer loop also. So, in the matrix form equivalently for this G, what we are getting is an m 1 by m 1, matrix. So, this being m 1 by m 1 matrix, we can write this thing in the form of a new matrix G 0. We are going to write down the G 0 representation soon, so this will be of the form W transpose G 0 w, this will be the matrix representation of this D F square term.

(Refer Slide Time: 38:17)



So that, I think, the first term, we have already got to which is nothing but W G. That is the first term representation and second term representation is W transpose G w where, G w is going to be, G o in this case is going to be like this, it is G of t 1 comma t 1, G of t 1 comma t 2 and last will be G of t 1 comma t m 1. And then, the second row will be G of t 2 comma t 1, because what we are doing is that the centers, there are m 1 centers, which we are writing along the columns. And there are m 1 number of inputs considering over here, so which is becoming G of t 2 to t 1, here this will be G of t 2, t 2 up to here G of t 2, t m 1.

And the last row will be G of t m 1, t 1 this will be G of t m 1, t 2 and these one will be G of t m 1, t m 1, so this will be our G 0 matrix representation Now; that means to say what, that means to say that we are having, now you see we had a term like this, d minus w j G, we were taking a square of that. So, essentially in the matrix representation, you remember that we had a matrix representation earlier also.

Now, essentially if we minimize the minimization of F starts; that is what we are attempting. Yields what, this will yields G transpose G plus lambda G 0, all of them are in the matrix. So, G transpose G plus lambda G 0, times the w vector is equal to G transpose d vector. I think this one, you should get here, what you can see is that, we are taking this square of this.

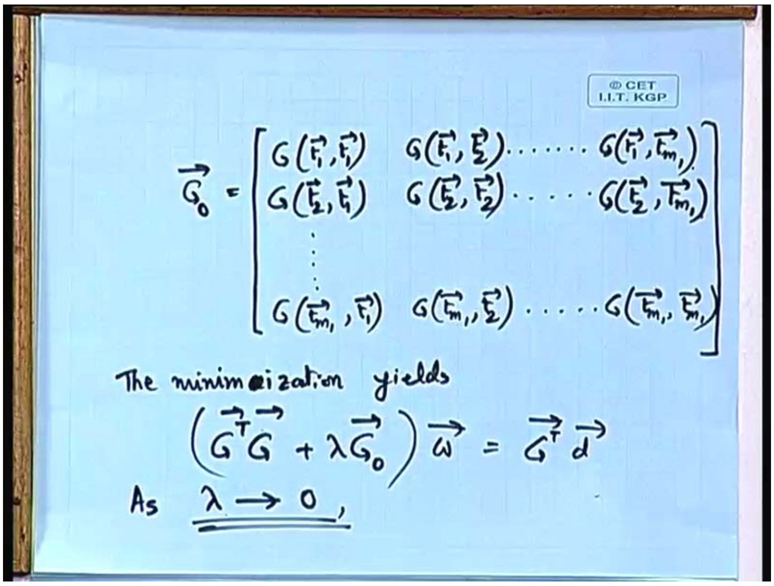
So, essentially this one becomes lambda square is

Student: ((Refer Time: 41:52))

W transpose, but actually D F square, we are already getting as W transpose G 0 w on differentiating, so now this is what we have got as the minimization equation. Now, what we have got is, that we have to solve for this w. Now, the solution of this we can obtain only in an approximation, only in approximated sense. Now, what happens is that, as the regularization parameter lambda approaches 0.

So, as lambda approach 0, then the weight vector becomes what, that we can find out from here, that the weight vector in accordance with this equation.

(Refer Slide Time: 43:08)



What we are showing as lambda tends to 0, the weight vector w is equal to G transpose G and we have to take the inverse of this, G transpose d, from this you see, as we make lambda attending to 0, this left hand side terms becomes G transpose G w. Now, this G transpose G should now go to the other side as inverse. So, it will be G transpose G inverse tines G transpose d.

So, this whole term G transpose G inverse G d, this could be consider to be, you see the dimensionality of this. This G transpose G is going to be of what dimension, this is going to be G transpose m 1 by m 1. So, G transpose G, again m 1 by m 1 and take the inverse of this and then, you are multiplying it by G, G is what, G is m 1 by N. G transpose m 1 by N and d is N dimensional vector and we get the m 1 dimensional solution to this problem.

So, w is of dimension m 1, so this equation should solve for w, now this term, what we had written as the products of this is written as the Pseudo inverse of the matrix G. So, this will be written as G plus, so defining that G plus is equal to G transpose G, inverse of that, this is the Pseudo inverse of matrix G. Now, that is under the condition that lambda tends to 0, we are making this terms, regularization term, very small.

Otherwise, there is a contribution from this lambda G 0 term also, actually speaking the strict solution for w will involve this also, but assuming that this contribution is very small. We can take it to be as G transpose G, so under that assumption, w is equal to pseudo inverse of G d. So, this is the solution, where we are taking the functional to be F start of x, instead of F of x.

So, instead of F of x, when we take F star of x and minimize it, up to m 1 number of terms, then the corresponding solution comes in this form. We had obtained, you remember, that we had obtained last time. The w solution was G inverse, not G inverse exactly; it was G plus lambda I, that inverse we have considered, whereas here, it is coming out to be on in the form of the Pseudo inverse.

Student: ((Refer Time: 46:51))

G t G, I had omitted this term, very correct, this whole thing is the Pseudo inverse, any questions pertaining to this, regularization function and the generalized RBF

Student: ((Refer Time: 47:29))

Yes, we are not considering this smoothing term, I think somebody already pointed out, that if we take this smoothing term, this is under condition that lambda is consider to be 0, lambda tends to 0, in that case, it is Pseudo inverse directly. Otherwise, if you are considering lambda, then also the solution is possible, then what happens, that w will become equal to this whole terms inverse.

This whole terms inverse times G t times d, where we should not have told this to be the pseudo inverse. We can tell this whole term as Pseudo inverse, only when lambda tends to 0, so thank you for today. So, we will have some discussion pertaining to RBF, that is to say, comparison between the radial basis function and multilayered Perceptron, which we will be covering in the coming class.

And then, we will be going over to the new chapter, which is the principal component analysis; that we intend to do from the next class on words,.

Thank you very much.