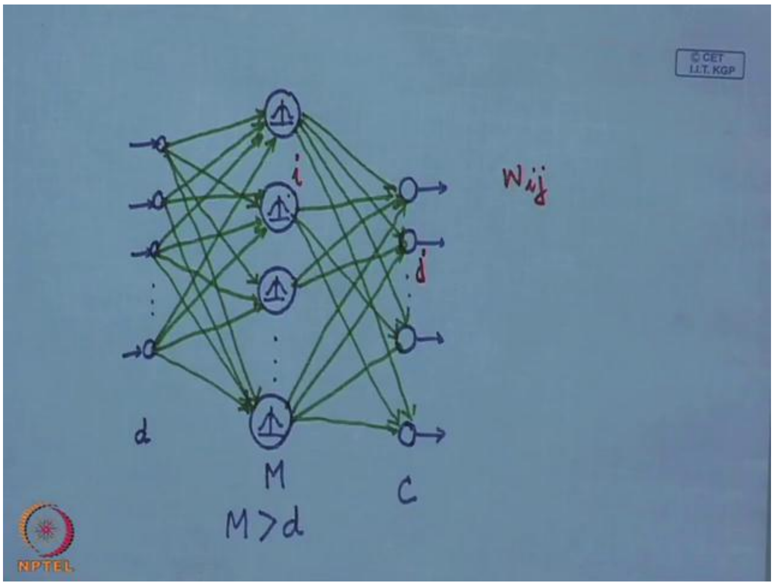
Pattern Recognition and Application
Prof. P.K Biswas
Department of Electrical and Electronics Communication Engineering
Indian Institute of Technology, Kharagpur

Lecture - 13
Probability Density Estimation (Contd.)

Hello, so in the last class we have started discussion on radial basis function, neutral network and we have seen that a radial basis function is neutral network consists of 3 layers. Basis one is the of course input layer and one of the three layer contain is outer layer and in between the input layer and output layer, we have a hidden layer. So, unlike in case of multi layer perception were can have one or more hidden layers in case of the radial basis function network we only one hidden basis and every neuron, in the hidden layer computes a radial basis function.

So, when I have the neurons in the hidden layer for every neuron which computes a radial basis functional, radial basis functional value for an input feature vector every radial basis function has got two parameters. One is called the receptor and other one which defines the spread of the radial basis function, so the architecture that we have is something like this.

(Refer Slide Time: 01:46)



We have one input layer, so the input layer contains a number of neurons and the number of neurons in the input neurons layered is same as the dimensional dimensionality. So, if the features vector is d in the hidden layer, I will have a number of notes and suppose the number of notes in the hidden layer is say M. So, as we discussed in our previous class that the purpose of the hidden layers notes is to project that the d dimensional notes into higher dimensional notes.

So, as I have a number of notes in the middle layer, so obviously this M is the number of note in the hidden layer, in the hidden layer is greater than the dimensional factor which is greater than d. As we said that every note in the hidden layer computes of this radial basis function like this. At the output layers which are basically the classifying rounds, I have the number of neurons or of the number of layer which is the same as the number of classes that we have. So, if we have c number of classes than the output layer I will have c number of neurons, so there we have seen C number of neuron, and C is the number of class in which the patterns are to be classified.

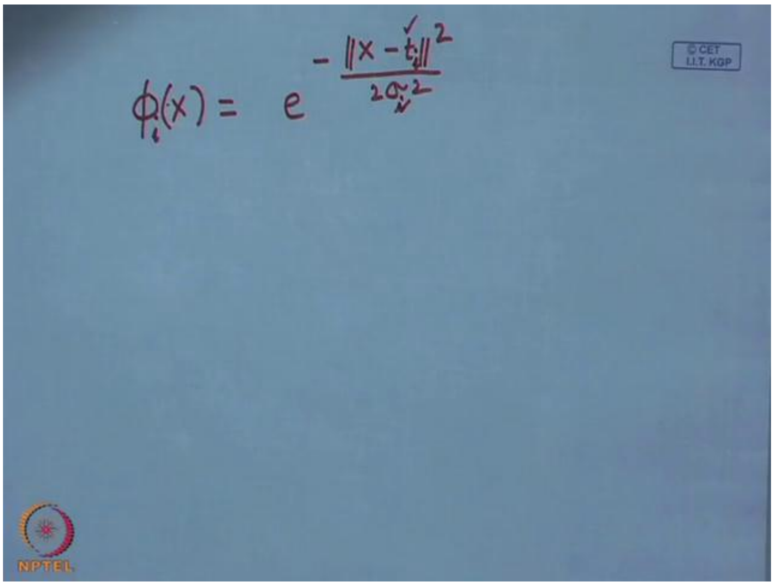
Then every note in the input layers connected is feeding input in every note hidden layer and output layer every layer note output layer from the hidden layer is connected to every note in the output layer. So, I have the connections which is something like this, so these are the connection form the input layer notes to the hidden layered notes because the propose of this connection is simply to forward the input factor to the nodes. In factor, the hidden layer we can assure that weight of this connection is equal to 1, and that is a difference with the connection from the hidden layer to the output layer notes because in every output layer note computes are linear to the output layer.

So, the connection from the output layer notes to the connection from hidden layer notes to the out layered notes is something like this. So, where we can see every hidden quote hidden there is connected to the j th note in the output layer through a connection with which is equal to W i j. So, because of this every node in the output layer computes, a liner combination of the outputs of the hidden layer and based on this the value of this linear combination the output layer notes decides to class the input vector should be classified.

Now, what can be done is these output notes can also impose on a liner function, to ensure that if a particular import feature vector belongs to class omega j. In the case only the output of the j th note will be equal to 1 and output of all other output notes will be equal to 0, similarly if feature vector input feature victor belongs to say class 1. Then only the output of the first note in the output layer will be equal to 1 and outputs of all other notes in output layer will be equal to 0. So, as we discussed in the previous class that search are radial basis function network and R B F network incorporates two type of learning.

One is we have to learn that for every node in the hidden layer because every node in the hidden layer represents a radial basis function what should be the receptor of that radial basis function. What should be the spade of that radial basis function, so if the radial basis function is a Gaussian function that is if it is something like this.

(Refer Slide Time: 08:27)



Say phi of X is equal to say e to the power minus X minus t square of this upon 2 sigma square, where t is the receptor and sigma which is the variance it decides that what the spade of the radial basis function is. So, every for every i-th radial basis function phi i X t i is the receptor and sigma i is the spade, so I have to know that what is the receptor for every radial basis function, and what is the spade of every radial basis function. So, this is one level of learning and the second level of learning is once through these radial basis functions are d dimensional feature vector is projected onto an M dimensional feature vector.

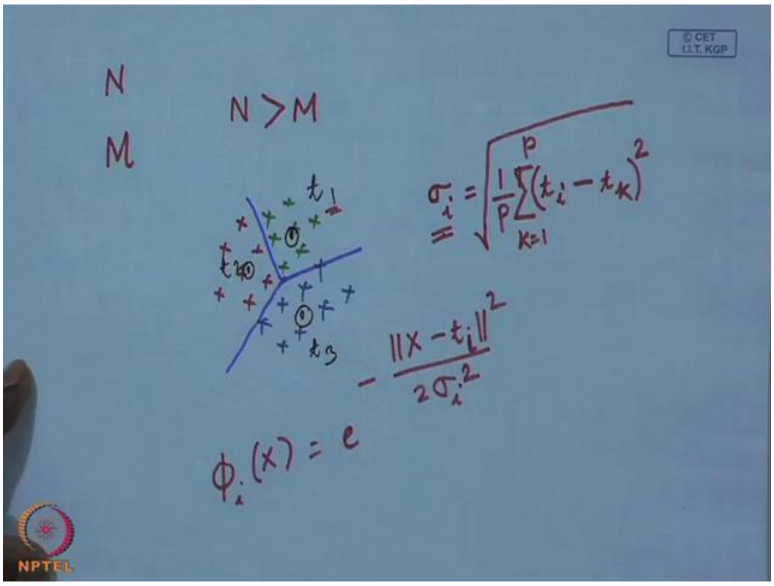
So, basically what we are doing is we are increasing the dimensionality of the feature vector, and as we have indicated in our last class that the purpose of increasing dimensionality is that. If the feature vectors are linearly non separable in the d dimensional space when we cast them in a hard dimensional space, then the possibility that they will be linearly separable in a hard dimensional space increases, and this possibility increases with the value of M.

So, as we increase that dimensionality more and more, the possibility of linear severability of the feature vectors also increases. The feature vectors in d dimensional space which are not linearly separable when I cast them into an M dimensional spaces and M is greater than d it is more likely that those feature vectors will be linearly separable in an M dimensional spaces. Once feature vectors are linearly separable in the M dimensional space then the linear combination of the outputs of the hidden layer is likely to give me the class belongingness.

Now, that liner combination is decided by the connection with from the hidden notes of the output lane notes, so we also have to learn that what should be the connection with W i-th note in the hidden layer to j-th from the node hidden layer, to the get nod in the output layer. So, this is the second leave of learning, so in the first level of learning for every radial basis function we try to learn what the deception is and what the separate of the radius function is.

So, for second level we try to learn what is the connection with from the hidden layer node to the output layer nodes and as we have discussed in the previous class that in souls with common method of learning. The radial basis of function is if you are given a slate which are of factors of the training purpose and suppose the value of M is 3. So, what we do is we partition or we cluster feature of factors into 3 number of clusters, so if we have M number of nodes in the hidden layer.

(Refer Slide Time: 12:26)

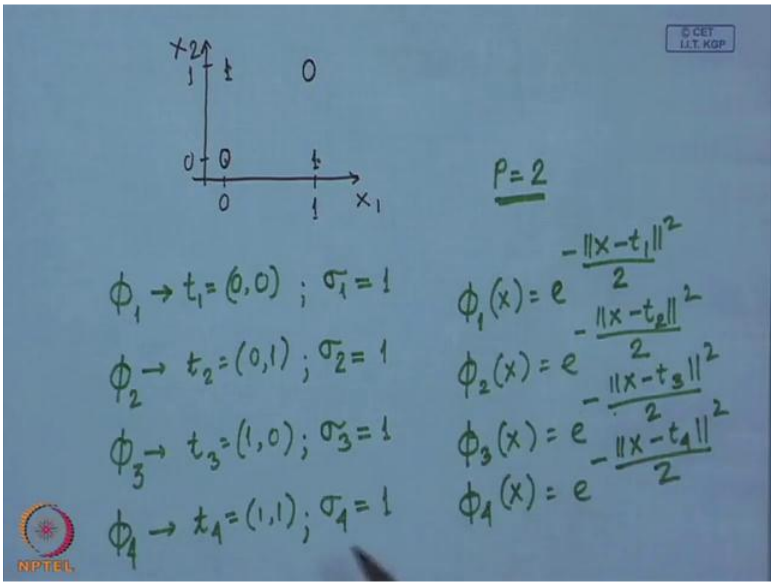


I have a number of N numbers of feature vectors, N number of feature vectors which are given for training purpose and I have M number of nodes of in the hidden layer. Obliviously, N has to be getter than M, otherwise clustering in N number of clusters which are into M clusters does not make sense, so I have more the number of cluster than the numbers of factors. So, I cluster this N number of vectors into M number of clusters and I can assume that centriod or mean of every cluster it represent the corresponding receptor.

So, if I take i-th cluster, the i-th cluster represents the receptor, i-th radial basis system, so this situation see if I have a set of which are vectors. So, these are the features factors belonging to different factors typically what I do is, I clusters this vectors 3 defined clusters every cluster center know represents receptor. So, this is one receptor and this is one receptor, this is one receptor, so this is a receptor t 1, this receptor t 2 and this receptor t 3. So, the first operation that we have to perform here is the coalescent feature of the vector and this plastering operation is we will discuss in detail in feature lectures. So, once I have these different receptors to find out what should be spades of a particular basis and function what you do is for height receptor i-th receptor find out p number of nearest neighbors or p number of nearest receptor. For this p number of receptors I compute what is the mean distance or root means quiet distance, so there are different possibilities. Take any value I can choose any value out of this p number of receptors, so what I do is the way I compute sigma, i for the i-th cluster for i-th radial basis function is I have t i which is the receptor for the i-th basis function.

Then I take p number of nearest receptor which are nearest to t i, so suppose one such receptor is t k, so what I do is I compute t i minus t k square. Take summation of this for k is equal to p, as I have p receptors one upon p of this and square root of this, so this defines the spread of i-th redial basis function. So, for every i-th redial basis function, t i and sigma i and once these two are known then radius function is phi of X is e to the part minus X minus t i square upon 2 upon sigma i square know. Let us see that by using this concept, whether I can make linear classifier using the radial basis function concept for the XOR problem and XOR is very common problem, which is used for illustrating such operation, so as we have said earlier.

(Refer Slide Time: 17:18)



If I take XOR function, I have a 2 dimension have been compounds X 1 and X 2, suppose this represent 0 this is X 1 2, 1, here I have X 2 is equal to 0. Here, I have X 2 is equal to 1, the value of the XOR function when 0, 0 is 0, 0 1 is value is 1, 1 0, value is 1 q and 1, 1 again the value is 0. So, you find that here i have two dimensional, binary feature factors and what I do is this 2 dimensional factor I want to cast into a 4 dimensional space by using four radial basis functions.

So, I have the radial basis functions phi 1, phi 2, phi 3 and phi 4, for phi 1 I choose t 1 is equal to say 0, 0, that is the receptor of the radian function. Similarly, for phi 1 I choose t

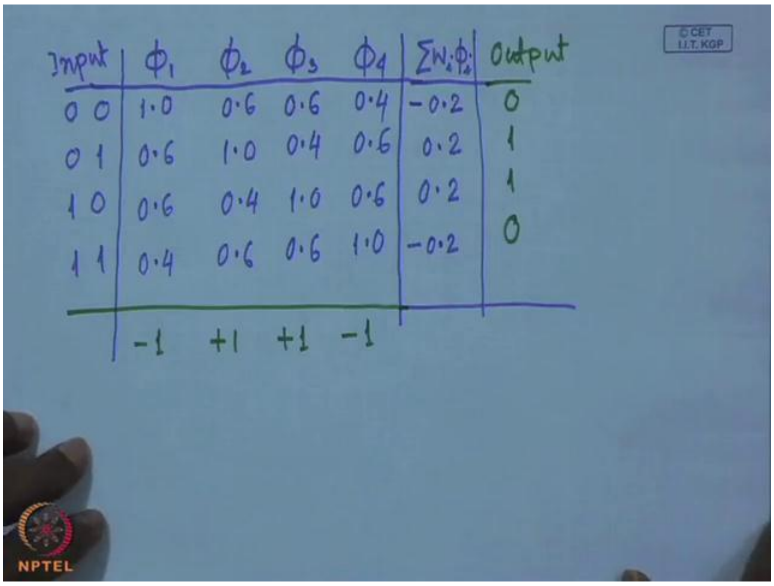
2 which is a 0 ,1 that is the receptor of the radial function phi 2, similarly the receptors of ideal function I can choose as t 3 is equal to 1, 0. For this I choose t 4 is equal to 1, 1, so these are the 4 receptors for 4 radius functions, next I have to choose the spread sigma 1 for the first aerial function.

I have choose sigma 2 for second aerial function, sigma 3 for third aerial function and sigma 4 for fourth radial function know for these for every receptor I have to find out p number of nearest receptors. Suppose I choose the value of p is equal to 2 know, here you find that for every receptor there are 3 neighbor 2 of a neighbors at a distance at 1 and one of the neighbors is at a distance of 1.4 that square root of 2. So, these is easily verify, here I have receptor 1 which is t 1, t 2 is at a distance of 1, t 3 is at a distance of 1, but t 4 is at distance of square root of 2 which is 1.4 or 1.14.

So, when I take p is equal to 2, I had to take two nearest neighbors both of them are at distance 1 and root means square a distance of this 2 distance will also be equal to 1. So, I have spade sigma 1 is equal to 1, I spade sigma 2 also equal to 1, I have spade sigma 3 also to 1 and I have spade sigma 4 that is equal to 1. So, I get phi 1 X which is of the form, e will be for minus X minus t one square of this upon W o sigma one that sigma 1 equal to 1, this will be equal to 2.

Similarly for phi 2 X, I will have a to the power minus X minus t 2 square of this upon 2 phi 3 X will be minus X minus t 3 square upon 2 and phi 4 X. So, if I compute these values for each of the factors vectors taking 0, 0 is one of the feature vector, 0, 1 has other feature factor, 1, 0 has another feature factor and 1, 1 has another factor vector the functional values will be something like this.

(Refer Slide Time: 22:43)



So, I put that form in a table, here I have input feature vector inputs are 0 0, 0 1, 1 0 and 1 1 and I have the phi function phi 1, phi 2, phi 3 and phi 4. So, when you input the feature vector 0 0 to phi 1 you find that your X is equal to t 1, so these expounds is equal to 0 which means phi 1 is equal to 1. So, this phi 1 X is over here this will be 1.0, similarly for phi 2 my X is 0 0, my t 2 is 0 1, so if I compute these phi 2 X you will find this phi 2, X will be equal to 0.6.

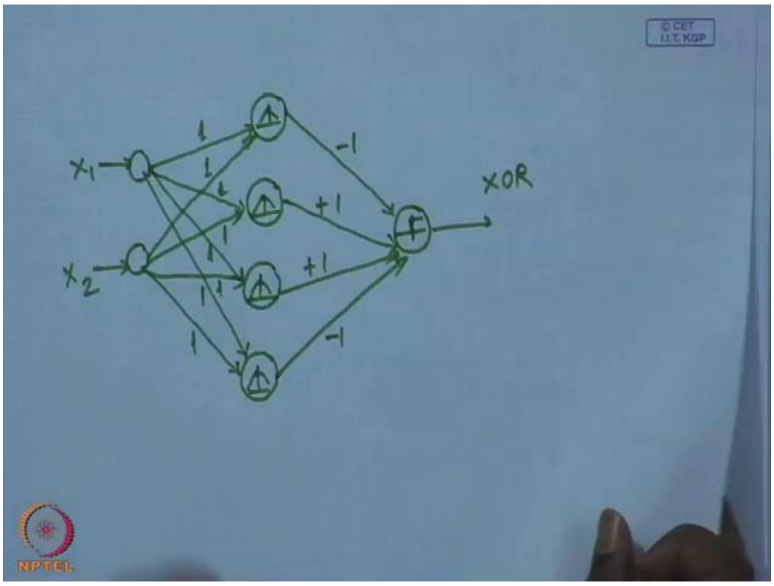
Similarly, I will put just values over here phi 3 X will also be 6 and phi 4 X is will be 0.4 in the input vector is 0 1, phi 1 X will be 0.6, phi 2 X will be 1.0, phi 3 X will be 0.4, phi 4 X will be 0.6. For 1 0 this is 0.6, this is 0.4, this is 1.0, this is 0.6 again and for the input vector 1 1, I have phi 1 is equal to 0.4, phi 2 X will be 0.6, phi 3 X will be 0.6 and phi 4 is that will be 1.0. So, you find that given our 2 dimension feature vector 0 0, this has been caused into a 4 dimensional factor vector where the compounds of this dimensional feature vector are 1.0, 0.6, 0.6 and 0.4.

Similarly, 0 1 there is a 2 dimensional input donal vector which caused into be 4 dimensional vector the compounds been 0.6, 1.0, 0.4 and 0.6. So, every input feature vector is a 2 dimensional feature vector, every 2 dimensional input vector is, vector is inverted to 4 dimensional factor vector by using 4 radial functions. So, if I take the linear combination of this and for linear combination for phi 1, if I give a weight of I give the weight for phi 1, I give the weight of minus 1. For phi 2 I give a weight of plus 1, for phi 3, I give a weight of plus 1, for phi 4, I give a weight of minus 1.

So, the function that I finally output at the output are node in the output layer will be phi 2 plus phi 3 minus phi 1 minus phi 4 and if I compute these let us see what are the values that I get. So, here I will weights sum of W i times phi i, i values from 1 to 4, so here it will be 0.6 plus 0.6 is 1.2 minus 1.4 this will be minus 0.2, here it be again 1.4 minus 1.2, so this is pulse 0.2.

Here, it will be 1.2 minus 1.4, so these is minus 0.2 and if I take a discussion that if the value is more than 0 the output will 1, if it is less than 0 the output will be 0, then the final output we have is. Here, I write out put this will be 0, this will be 1, this will be 1 and this will be 0, so which is nothing but the X of function output, so over here the architecture of the radial basis function that we have used is.

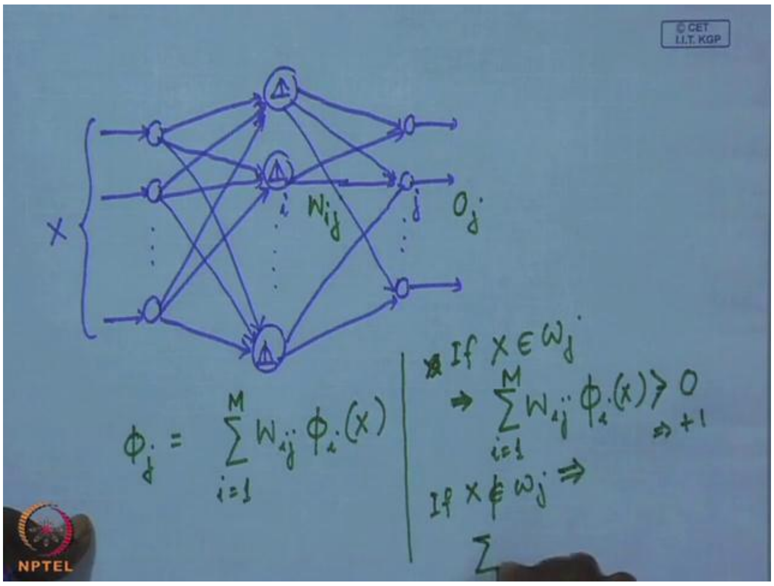
(Refer Slide Time: 28:08)



We had two input layered nodes with X 1 is spade to one node, X 2 is spade to another node, I had 4 nodes in the hidden layer which computes the radial basis function and I had 1 node in output layer which I can say that it is finding out. It is a non linear operator or a threshold operator the connection that this quite where each of this connection have a connection weight is equal to 1 and over here these connections. As you can see over here phi 1 to output layer minus 1, phi 1 to output layer has connection of plus 1, phi 3 to output layer nod has connection of plus 1, phi 4 layer nod is connection with of minus 1.

So, here the connection minus 1 plus 1 plus 1 minus 1 and this output actually gives me the XOR function, so this example clearly shows that by casting the 2 dimensional feature vectors into 4 dimensional feature vector I can implement the X of function. Using a liner network or a single layer preceptor because this part is nothing but a single layer preceptor, now let us theoretically try to find out. Try to find an expression for the training of the output layer or how do I find out this connection weights, so in general I have a network something like this.

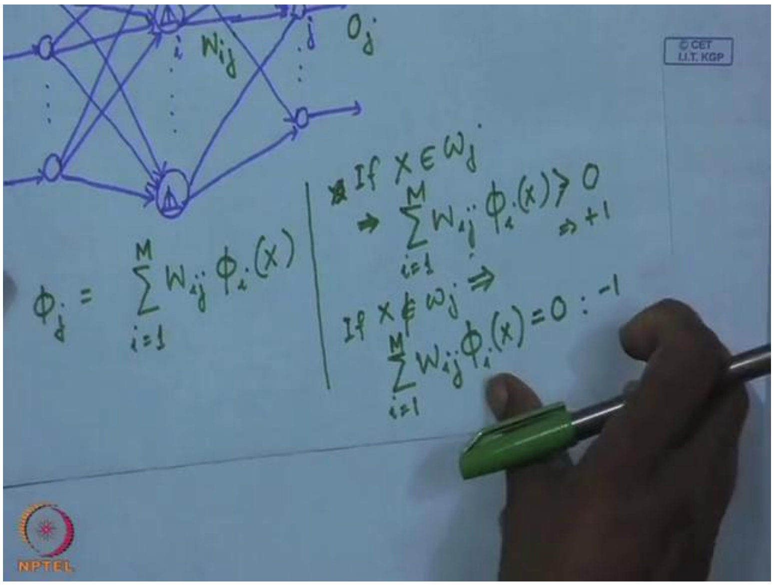
(Refer Slide Time: 31:02)



I have a set of input layers, set of input layer nodes I have a set of hidden layer nodes and I have a set of output layer nodes, the feature vector is fed to the input layer nodes. So, here I fed X which is fed to the hidden layer nodes through connection weights which is 1 and outputs of this hidden layer nodes. These are my radial basis functions outputs of the hidden layer nodes are connected to the output layer nodes, so like this and I take the output from every output layer node. So, if my input feature victor X belongs to say i-th class then output of the i-th output note will have a high value likely to be 1 and outputs of all other output layer node will have a low value likely to be 0.

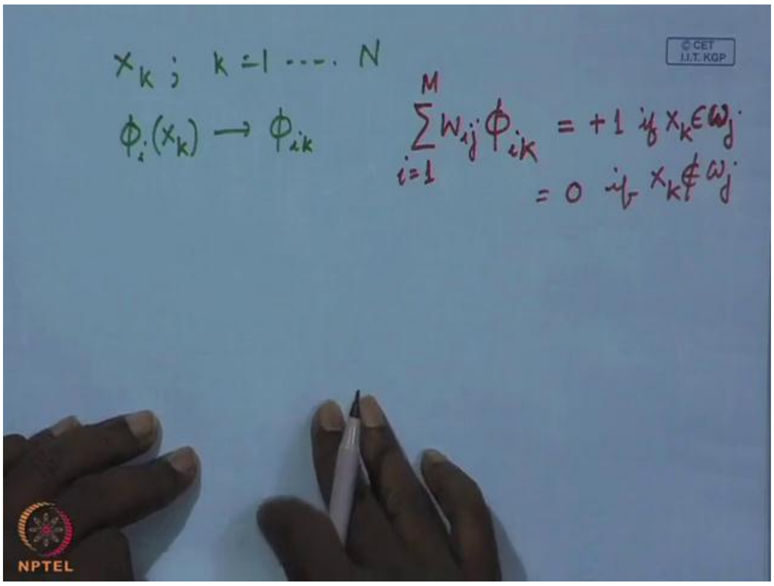
Now, I assume that the i-th node in the input layer is connected to the j-th node of the output layer through a connection with say W i j. So, given this if I say the output of the i-th, j-th node is O j i will have O j is equal to sum of W i j times phi i X for an input vector X and this summation I have to compute over all nodes in the hidden layer. So, here I will have this summation has to be computed over i is equal to one to M as I have M number of nodes in the hidden layer and naturally over here if this feature victor X. So, I will write if X belongs to class omega j then I have to have some of W i j times phi i X, i is equal to 1 to M this has to be equal. This must be greater than 0, I will put this as plus 1 and if X does not belong to omega j that indicates that sum of.

(Refer Slide Time: 34:41)



W i j into phi i X i having from 1 to M that must be equal to 0 or I can also put it as minus 1, so let us assume that if X belongs to plus omega j, W i j-th times phi i, phi i X that has to be equal to plus 1. If X does not belong to omega j this has to be 0 and that is what has to be the output from the j-th node in the output layer, now taking this. Now, I can go for training of the output layer that means I have to find out what should be the values of this W i j. Now, if I compute only the connections weights, if I right now consider only the connection weights which are connected to the j-th node in the output layer then for every vector X k, suppose I have N number of vectors.

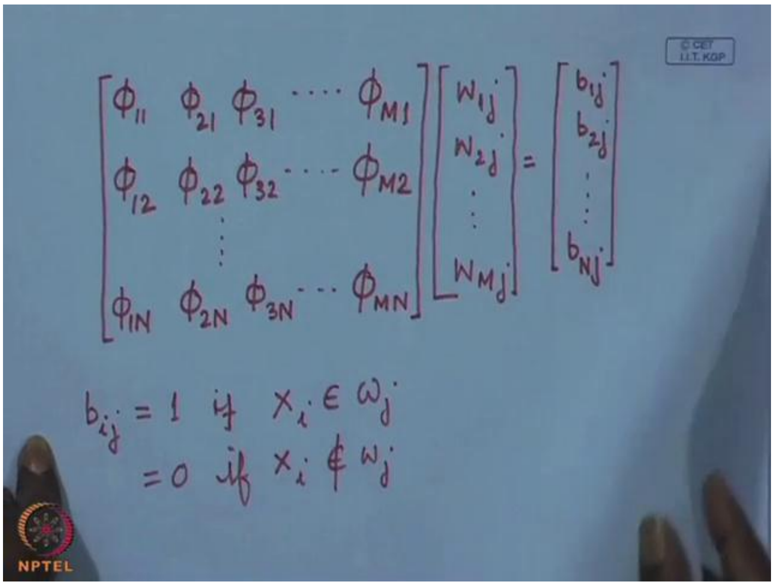
(Refer Slide Time: 35:57)



So, I have vectors X k for k varying from one to M, I have capital N number of input vectors which are given for training purpose or for learning as we are supervised learning then phi i of X k, for simplicity i will write this as phi i k. Now, by using this I can as I said that sum of phi i X k into W i j k, so my condition is if you remember this one if you remember this 1.

So, sum of W i j into phi i k for i varying from 1 to M this has to be equal to plus one if X k belongs to plus omega j if X k belongs to omega j and this has to be 0 if X k does not belong to omega j. So, this is the output that I expect, so for X k I have for every X k, I have such a kind of linear equation that this summation will be either plus 1 or 0 and all those N number of equations, now I can write in the form of a matrix, so in the matrix form this can be written as.

(Refer Slide Time: 38:02)

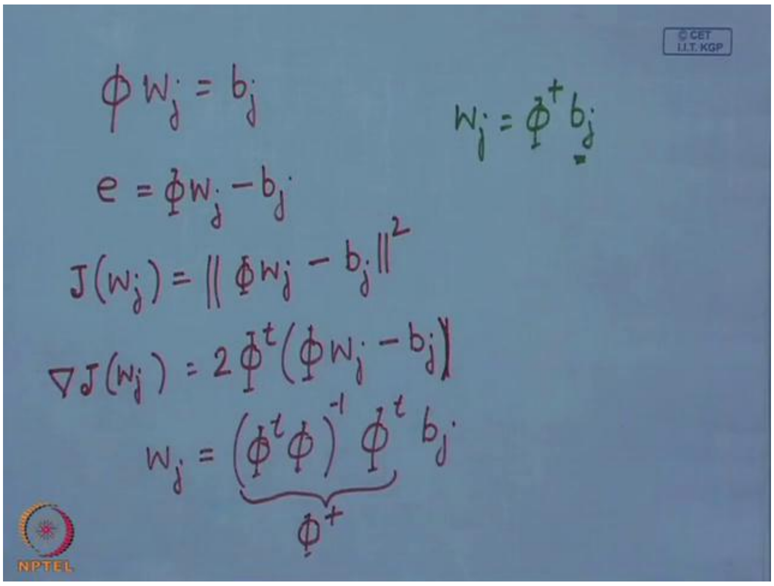


Let me write the matrix equation that phi 1 1 which means phi 1 X 1 phi 2 1 that is phi 2 of X 1, phi 3 1, phi M 1 this means phi M of X 1. Similarly, phi 1 2 which means phi 1 of X 2, phi 2 2, phi 3 2 up to phi M 2 and as I have M number of samples for training, so I will have phi 1 N, phi 2 N, phi 3 N up to phi M N which indicates phi M of X N into W 1 j, W 2 j up to W M j. So, if find the what it computes W 1 j times phi 1 1 plus W 2 j phi 2 1 continue like this W M j phi M 1 that is for the first input vector X 1.

Whatever is the output of individual middle layer nodes or hidden layer nodes this equitation simply makes a linear combination of outputs hidden layer nodes for the input vector which is one or X 1. So, these has to be equal to again i output in form of vector b 1 j, b 2 j up to b N j where every b i j will be equal to 1 if the corresponding X i belongs to plus omega j and that will be equal to 0 if the corresponding X i does not belong to 1 plus omega j.

So, every b i j will assume a binary value either 0 or 1, so this b i j will be equal to 1 if X i the corresponding input vector X i belongs to class omega j the j-th class or it will be equal to 0 if X i does not belong to omega j, so this the kind of situation I have and these whole expression this matrix equation I can write in a short form.

(Refer Slide Time: 41:38)



That is phi W j is equal to b j and this phi is this matrix and W j is this weight vectors which are connected to the output layer j and b j is the output of the j note in the output layer which is represented in the vector line like this for different input vectors. So, if the network is properly trained that is all W i j has got the trained value then this equation should be satisfied. But, what we are trying to do is we are trying to train the networks we are to set the weights W j, so it cannot expect that this equation will be satisfied essentially. So, if the equation is not, if this equality is not satisfied then what I can do is I can define a error e which is nothing but phi W j minus b j and, now training involve adaptation of the W j.

So, that this error can be minimized see in order to do that as we have done earlier or mean square error optimization for square error technique for classified learning or classified training I can also define. Here, a function j of W j which is given by phi W j minus b j norm of this and then I take gradient to with respected W j, so grade of j W j which will be simply 2 phi transpose into phi W j minus b j. By equating this to 0, what we get is W j is equal to phi transpose phi inverse of this into phi transports b j, and as we have seen earlier this phi transpose phi inverse into phi transposes.

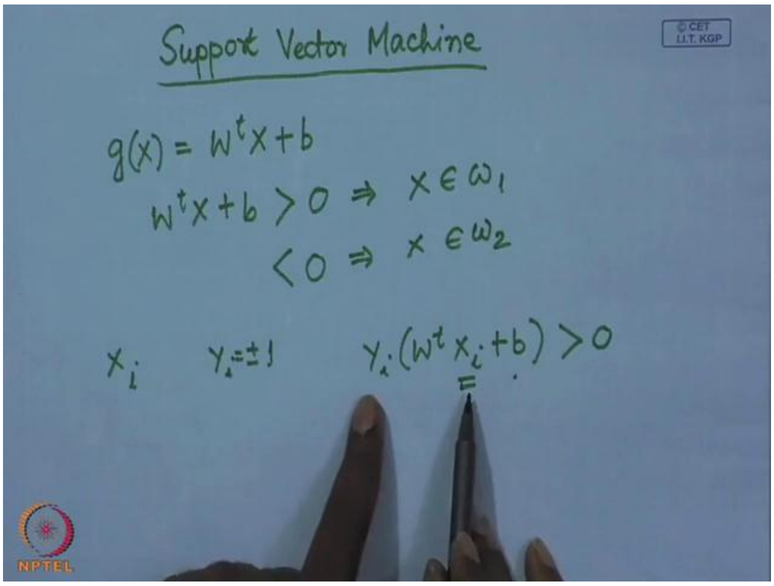
This is what is called should inverse and that is represented as phi plus so we have W j by this should inwards technique we have this W j is equal to b j, b j is defined every compounds either 1 or 0. It will be equal to 1 if the corresponding input features belongs to gel and compounds will be equal to 0 if the corresponding that does not belong to class omega g. So, I have this vector b j phi actually indicates that what should be output hidden layered nodes by the vector from that i compute what is matrix. So, once I have matrix phi and I have this b j, I can compute what will be the connection for different nodes layer to the j-th output layer.

This if I do for every output layer I can compute what is the connection width from different outputs of the hidden layer nodes for different output layer nodes. That is what completes my training of R B F L nitrite work and the labial neutralize will ready for classification know if you compare neutralize work with your against the multi precept in fine. The training of the R B F neural network is faster than the training preceptors because in case of preceptors the training is done which takes lot number of introduction. So, the training of the R B F neuron will be faster than training of the multi layer preceptor the second advantage is that I can easily interpret what is the meaning or what is the function of every node hidden.

Here, which is difficult in case of multiply layer perception, I cannot easily interpret the role of different correction in the hidden layer in case of multi layer perception and not only that I also cannot easily decide that what should the number of layers. What should be the number of hidden nodes in every layer, so those are the difficulties in the multi layer preceptors which is not there in case of are B F nitrides however hi p f network as a disadvantage that though the training is faster.

But, you find that the classification takes more time in case of erective network then in case of N p because in case R B F network every nod in the he hidden layer has to compute the radii basis functional value for the input network. So, the clarification in case of classification takes more time than the classification time in case of multiplier extra, so with is we come to a consolation on the radial basis function neural network know over here I will just make classification about another kind of classifier which is called a support vector mission.

(Refer Slide Time: 48:37)



So, I will briefly discuss support vector machine so support vector machine is another type of linear classifier, so if you remember what we discussed in last case of a linear classifier that given to classification. We have said that I can define a discriminating function say g of X which is of the form say W terms pose X plus b and we have said in case of linear discriminator. In case if this g is factor W transports X class b this is greater than 0 that indicates that X pulse of omega 1, and if this is less than 0 then which feature vector X pulse omega 2.

So, here find for the classification purpose the actual value of g X is not really very important, but what is important what is the sign of g X, if the sign is positive I infirm that X belongs to mega 1. If the sign is negative i infer that X belongs to plus omega 2, so over here with every X if with every X i indicted number Y i, Y can be either plus 1 or minus 1. In that case this Y i times W transpose X i pulse b it will be always greaten 0 if the sample X i is properly classified which quiet obvious because if I say that Y i is equal to pulse 1 for a sample X i belongs to omega 1.

For example, which is belongs to omega one this transports X i is greater than 0, Y i is also positive, so Y will obliviously greater than 0 if X i belongs to omega 2 then W transport W X i, it will be less than 0 and for that I set y i e. So, minus 1 times W transposes X i will obviously greater than 0 and this is a concept that actually we have used when we have discussed about on the percept on criteria or design the linear classifier. That is for every feature vector belonging to class 0, we have negative feature vector before we try to design the classifier.

So, the every feature vector irrespective of whether the feature vector belongs to plus omega 1 or the feature vector belong to class omega 2 my discrimination function value will always be positive, if the feature vector is correctly classified. So, that is true if the feature vector belong to class omega 1 or if the vector belongs 2 vector because the feature vector belonging to omega 2 before raying to design the classifier. We have negated vector, so we will discuss about support vector machine more in details in our next class.

Thank you.