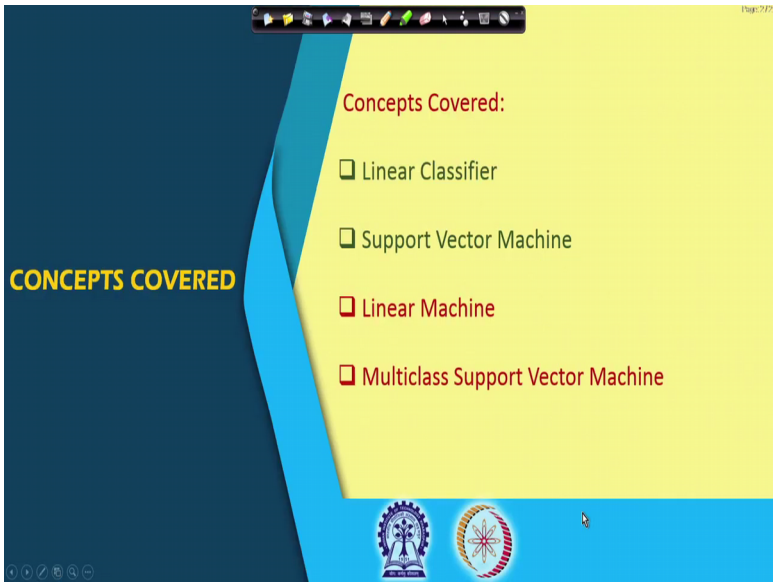
Deep Learning
Prof. Prabir Kumar Biswas
Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology, Kharagpur

Lecture - 14
Multiclass Support Vector Machine - I

Welcome, to the NPTEL online certification course on Deep Learning. You just try to recapitulate what we have done in the previous class.

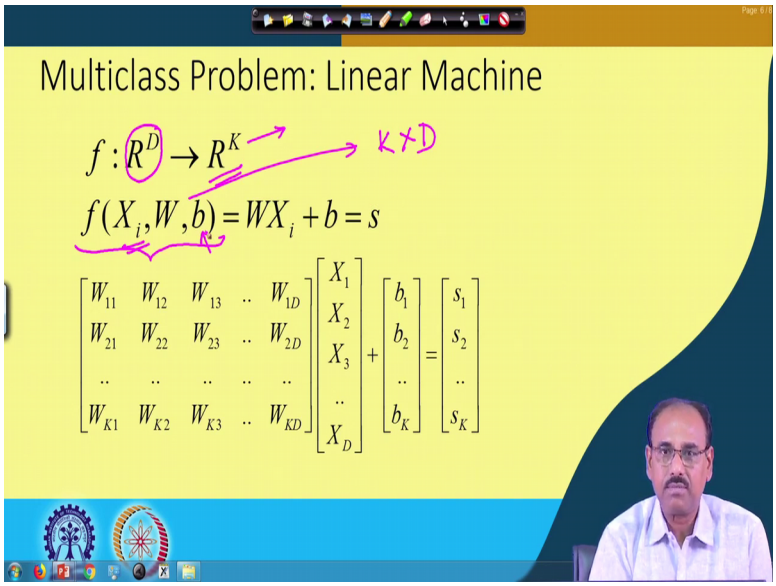
(Refer Slide Time: 00:39)



In the previous class, we have talked about the linear classifier and, then we had moved on to the linear machine. In today's class we will continue with the Linear Machine and then we will move to what is known as Multiclass Support Vector Machine.

So, as you remember in the previous class that a linear classifier when I extend that to a multi class problem because linear classifier is usually in case of a two class problem where we try to find out what is the boundary between the two different class and we try to design that boundary using the feature vectors or the training vectors given from the two different classes. When it comes to multi class problem then what we have a linear machine.

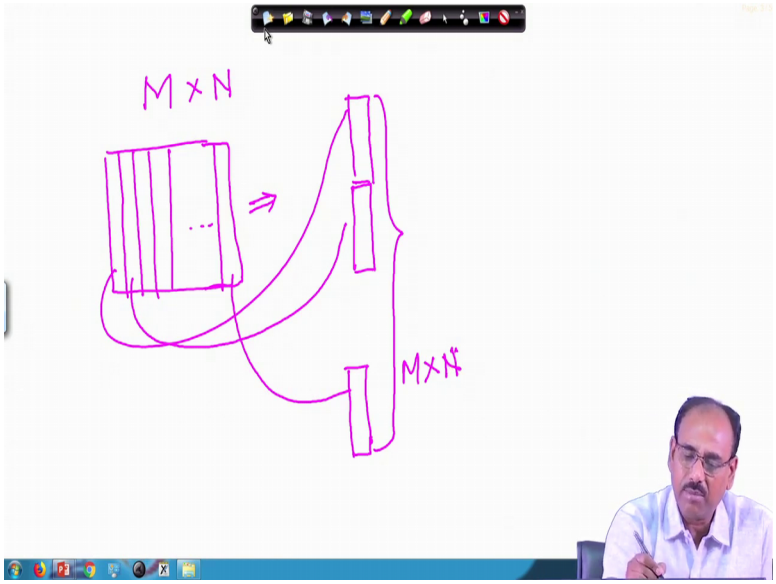
(Refer Slide Time: 01:33)



So, if you remember from the previous class a linear machine tries to map a D dimensional feature vector say R D to a K-dimensional feature vector which is known as which is R K. So, this R K is what is known as the score vector that is given a feature vector X i when the linear machine maps that vector to a K dimensional score vector s every component of s tells us that what is the score for that particular category the corresponding to the component of s for the given input vector X i. So, the first question is that how do we get this D-dimensional vector R D.

Let us take a very simple case that we have talked about generation of the feature vectors in conventional machine learning techniques, where the feature vectors are interrupted and they can be generated using the properties of the boundary of the object or using the properties or regional properties which contains the color information texture information and intensity information and all that, but when we talk about a deep learning we do not depend upon the handcrafted feature vectors rather we want that the machine will learn the features as well based on which it will classify or it will understand the data.

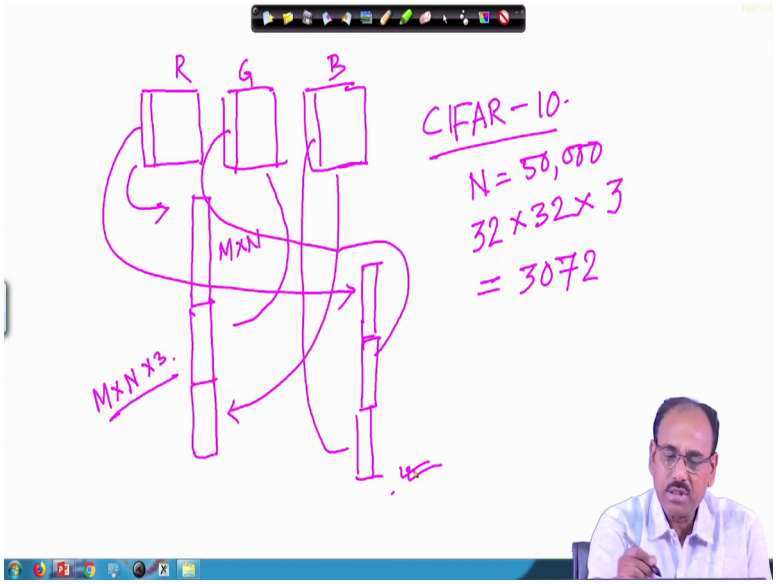
(Refer Slide Time: 03:31)



So, for specific example when we have image as the input data, so, suppose we have an image of size say M by N; that means, M number of rows and N number of columns so, which is a matrix of integer numbers. So, this is say an N M by N image which will have N number of columns and M number of rows. So, the though way I can convert this into a feature vector is first you take this column and form a part of the vector, then you concatenate the second column that makes another part of the vector like this you continue the last column forms the last portion of the vector.

So, when I have an image of size M by N having total number of pixels which is M into N I have a feature vector of a generate a feature vector from this image which has got M into N number of components. So, this is an M into N dimensional vector.

(Refer Slide Time: 04:49)



Likewise if I have a color image then in color image we have three different planes I have red plane, I have green plane and I have blue plane each of these color planes. Let us assume have got M by N number of pixels having M number of rows and N number of columns. So, from first R component R plane I generate a vector having M into N number of elements. Concatenate with that the vector generated from the green component having again M into N number of elements and then you concatenate the similar vectors which are generated from the blue component again having M into N number of components.

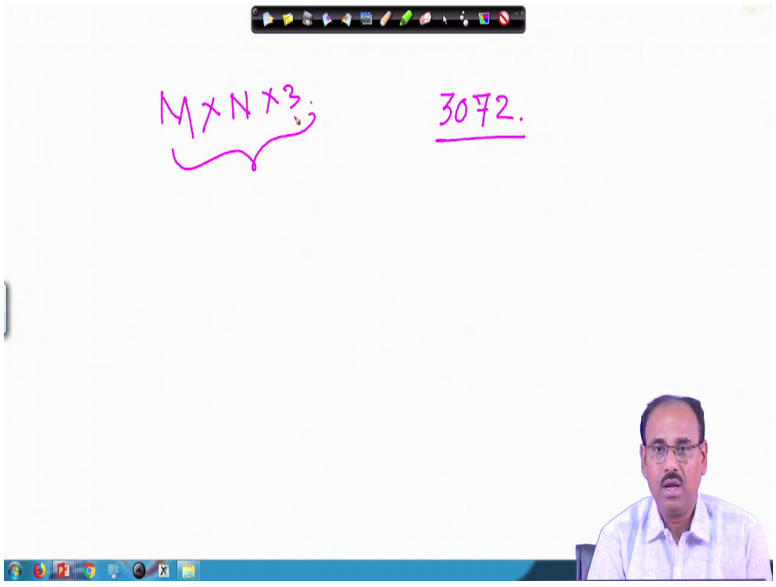
So, the total number of elements in this feature vector now becomes M into N into 3 this is the total number of elements in this feature vector. So, we were talking about CIFAR-10 database we have studies in the previous class and in CIFAR-10 database we have assumed that there are N is equal to say 50,000 images. Each image is of size 32 by 32 pixels and those pink color images there are three planes; red, green and blue for each of the images.

So, if I convert this into a feature vector of this form every image will be converted into a column vector having say this is 32 by 32 into 3. So, that is equal to 3072 number of elements; so, one element corresponding to every pixel. So, when I convert an image into a vector of this form this vectorization the way I have shown is not the only option available. What I can do is first I can have one column of component R, so, using this

column I had make a part of the feature vector, then you take one column of green make another part of the feature vector, then you take another column from B or the blue component make another part of the feature vector.

And this will continue for every columns taken from red, green and blue components. So, this can be another way of making converting a colored image into a vector form.

(Refer Slide Time: 07:45)

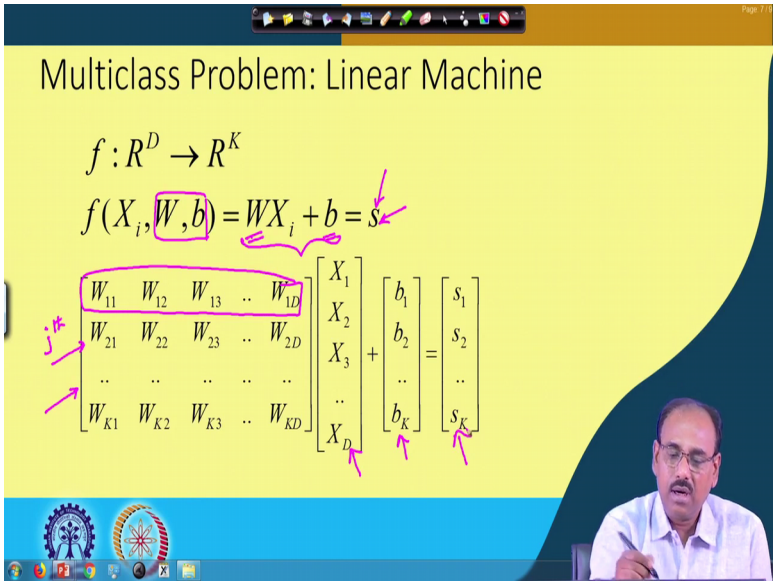


And, another thing you notice that one I have once I have converted an image into a vector of this form so, this vector is of dimension M by N by 3 where the images are of size M by N that the images are color means for CIFAR database this will be having 3072 number of components. So, that means, I am defining a space or 3072 dimensional space and every image is now represented by a point or by vector into that 3072 dimensional space and this tells every image will be defined by vector or it will be represented by a point into M into N into 3-dimensional space. So, that is what the vector representation of an image.

So, once I have such a vector representation so, here this is what I was telling that I have this d dimensional vector which is represented which is generated out of an image and the linear machine maps this D-dimensional vector into a K-dimensional vector, where K is the number of categories or the number of classes that we have and if I expand this expression this expression can be written in this form that f X i, W, b where X i is my input vector, W is a matrix of dimension K by D having K number of rows and D

number of columns and b is a bias vector having again K number of elements. So, this is a K dimensional bias vector.

(Refer Slide Time: 09:51)

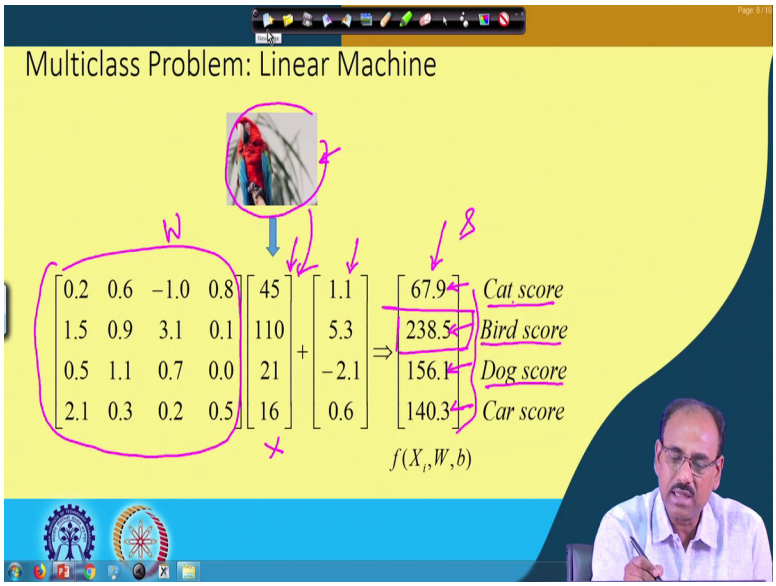


This W and b they are the parameters of the linear machine. They define the linear machine.

So, this functional form f X i, W, b is nothing but W X i plus b where W as I said that it is a parameter matrix or weight matrix, and b is the bias vector. So, W and b taken together and they are parameters of the linear machine and this operation W X i by b gives me a column vector again of dimension K, where this s a K-dimensional vector is what is my score function. So, if I put it in an expanded matrix form this is the matrix expression that I get.

So, here every row of this matrix W represents classifier of a particular class. So, if I take j th row of this matrix W this represents the classifier of the j th class and this is my input vector X, there is the bias vector and after computation I get the score function K right.

(Refer Slide Time: 11:15)

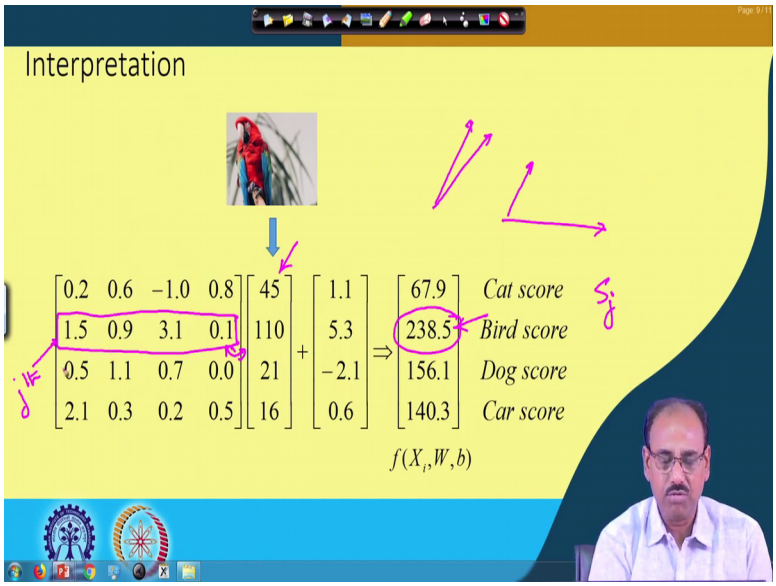


So, now let us take an example to see what it actually means. Suppose, I have somehow I have obtained this which matrix W, this is the vector which is generated out of an image let us assume that this vector for simplicity is a vector of dimension 4. So, I have got 4 elements in this particular vector and there are 4 number of categories as well in this particular example. So, my bias vector is again a D-dimensional bias vector. So, this vector X you are getting from this particular image using the technique that we have just discussed and after this equation I get this score vector s which is having again 4 number of elements because I have got four categories.

And, you find that each of this each element of this score vector s gives you the score for a particular class. Here the first element let us assume that the first category is cat category; so, the first score corresponds to a cat score. The second category is bird category in this particular case; so, the second score corresponds to bird. The third score corresponds to dog and so on and, here as my input vector is generated from a bird so, you find that the bird score is maximum. It is maximum of all other scores ok.

So, my for my classification purpose whenever an in unknown input vector is presented to the classifier, the classifier will generate or the linear machine will generate a score vector and in that score vector whichever component comes out to be maximum I have to classify that input vector to that corresponding category.

(Refer Slide Time: 13:19)



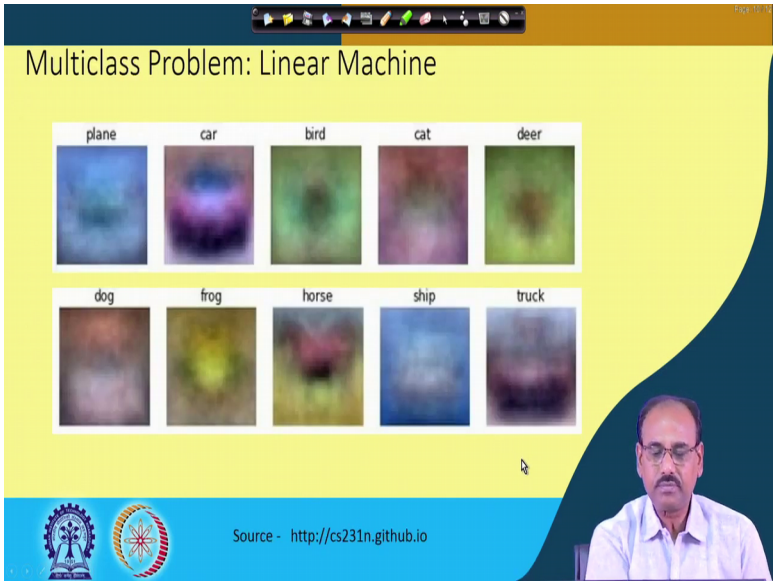
Now, here comes the interpretation of the rows of weight matrix W. You find that effectively what I am doing is this matrix calculation is doing is that it is taking the dot product of the input vector it is taking dot product of the input vector with say j th row vector of weight matrix W, suppose this is the j th row. So, it is dot taking the dot product of the input vector with the vector corresponding to the j- th row of the weight matrix or the parameter matrix and it is generating the j th component of the score vector s.

And, as we said that this being a vector that is my input image represented as a vector or as a point in D-dimensional space and this j th row of matrix W is also a point in the D-dimensional space and I am taking the dot product of these two and you know that the dot product of two vectors tells you what is the degree of similarity between the two vectors. So, if the two vectors are very similar say if I have one vector in this and another vector like this, the dot product of these two will be quite high; whereas, if I have one vector here and another vector here which are widely different that dot product of these two will be quite low.

So, this j th component of my score vector or s j in other words tells me that what is the similarity between the input vector which I want to classify and the vector corresponding to j th row of the weight matrix W. So, having this interpretation of the weight matrix I can say that every row of the weight matrix is nothing but a template of the

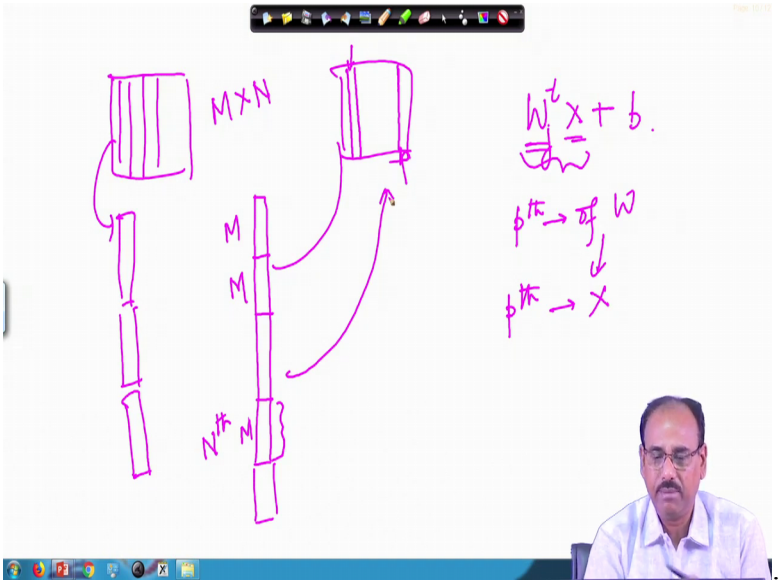
corresponding category. So, the j th row is represents a template of the j th category the K th row represents a template of the K th category.

(Refer Slide Time: 16:03)



So, if you again coming to the example of CIFAR database we also mentioned in our previous class that the CIFAR database contains images of 10 different categories like plane, car, bird, cat, sheep and so on and after training the rows of the weight vector the rows of the weight matrix as I said that they represent the templates of different categories; so, if I convert those rows into the to the form of images every image will be a template of the corresponding class of the corresponding category.

(Refer Slide Time: 17:03)



So, how do you convert this weight vector into an image? So, it is just the reverse process that as we have shown in the previous example that the way we convert an image into a vector is that I have an image. I take every column of this image one by one, concatenate those columns to get the vectors. So, this is how I form a vector from an image.

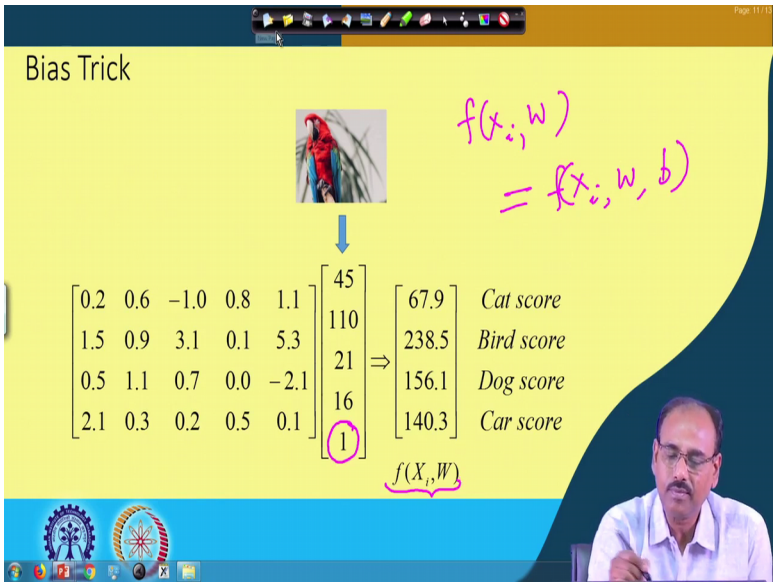
Similarly, if I have a vector how do we convert this into an image. So, if you look at the way we have done this multiplication that is when we have done W j transpose X plus say b where this is a dot product of the vector corresponding to j th row of the weight matrix with the input vector X. What we have done over here is that every p th component of W p th component of W is multiplied with p th component of my input vector X; that means, p th component of my weight vector W of the j th row corresponds to p th pixel in the vector representation of my input image.

So, as I have converted an image into a vector by unfolding the columns and concatenating them together to form a vector of dimension D when I do the reverse process that is given a vector I want to form a matrix out of it. So, first you take the first M number of elements of this matrix my image was I assumed to be of size M by N. So, I take first N number of elements of the matrix of this vector and form the first column of the image; take the second N number of elements of this vector and use that to form second column of the image.

Similarly, at the end I have M th N number of elements of the vector and use that to form the last column of the matrix, it should be the other way I have M number of rows. So, instead of N number of elements I have to take M number of elements. So, here I take first M components of the of the vector from the first column of the matrix then you take the second M number of elements of the vector from the second column and in the same manner you take the N th M number of elements of this vector from the last column.

If I have color images obviously, I will have other color components. So, accordingly I form other color planes of the input image and, this is how I convert the vector W in the form of an image. So, this is what we have the way those vectors can be converted into an image gives me the template. So, you find that these are the templates corresponding to different classes.

(Refer Slide Time: 21:11)

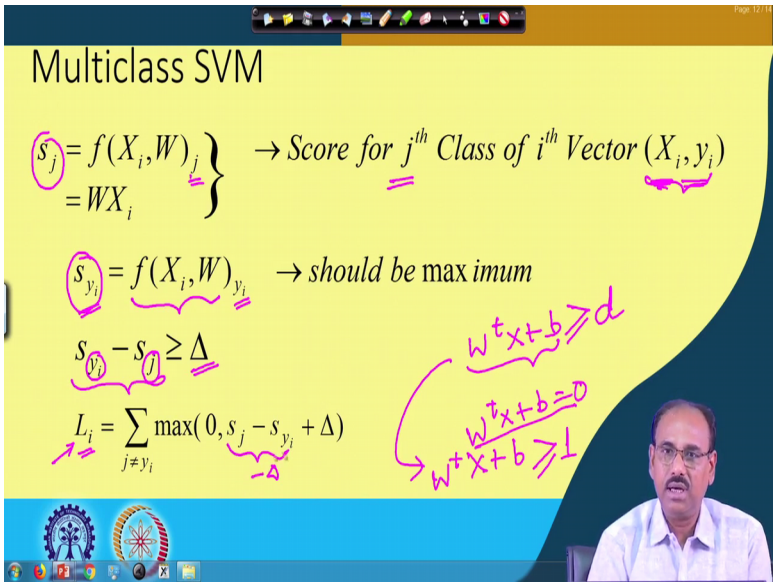


The next we come to the case of we have a bias vector. It is also possible to include that bias vector in the weight matrix itself. So, for that what I have to do is, I have to simply increase the number of columns of the weight matrix by 1 and that additional column of the last column is actually the vector column vector corresponding to the bias vector and for doing that the modification that I have to do in my vector data vector is that I have to add an additional element and make that equal to 1.

So, this gives me the same matrix equation only difference is now I do not have the bias vector separately, but it is included in the weight matrix itself and accordingly the

function that I compute is f X i, W where W includes the bias vector v b and you can verify that f X i, W is same as f X i, W b when the bias vector was kept separate. So, both these computations are same. So, accordingly the score vector that you get the score vector will also be same.

(Refer Slide Time: 22:09)



So, what I get is I get for every vector I compute the function as given by the linear machine and the linear machine gives me a score vector which is s. So, if I take s j or the j th component of the score vector as is shown over here. So, this s j which is the j th component of the score vector which is nothing, but when I do this matrix multiplication WX i and taken the j th component of that which is nothing but the score for j th class for the i th vector X i and i th vector X i is actually given in the form X i, y i indicating that this X i belongs to category y i.

You remember that what we are talking about are all training vectors; that means, for the training vectors we know the category to which the training vector belongs. So, I know that the training vector X i belongs to category y i and given this to obtain the score for y i th category for the vector X i have to consider I have to look at what is the y i th component of the score vector and y i th component of the score vector as before is nothing but whatever I get by this matrix multiplication and take the y i th component of that and because we know that this X i belongs to category y i; so, this s y i that is y i th

component of my score vector X must be maximum among all other components of the score vector because this X i belongs to class y i.

So, the next question is should we be satisfied just with the condition that s y i is maximum or I want that this s y i should be more than all other components at least by some margin delta so that I can say with confidence that whatever classification I have got that is correct. You remember with what we did in case of support vector machine that in support vector machine your W transpose X plus some bias b we assumed that this have to be greater than some minimum threshold d.

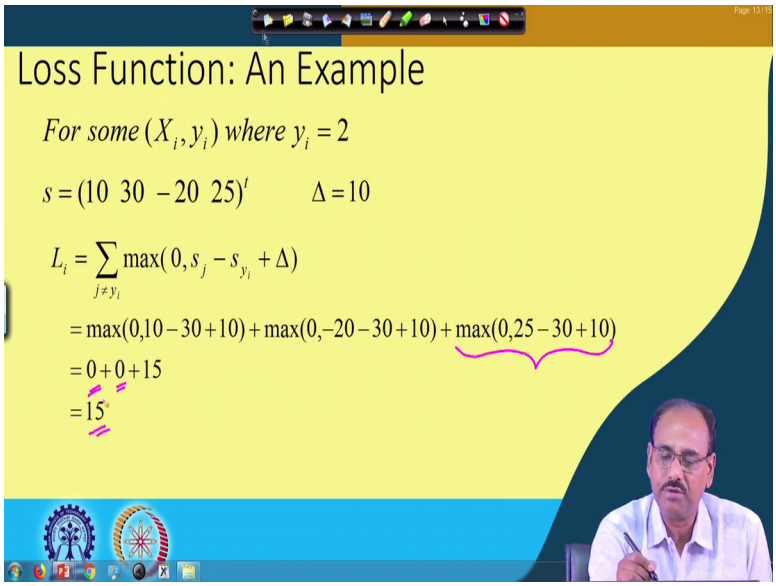
And, what is this W transpose X plus b; this tells you that an idea of what is the distance of X vector X from the plane W transpose X plus b equal to 0 and with normalization this we can always write in the form W transpose X plus b have to be greater than or equal to 1 after normalization we can always do that. So, this says that from this boundary W transpose X plus b equal to 0, the normalized distance of my training vectors must be greater than or equal to 1 and that is the margin.

Similarly, in this case to have confidence over the result that we get the classification result that we get I should have I should impose that the score for class omega i must be greater than score for any other class j at least by margin delta and accordingly I can define a loss function L i which is given by 0 maximum of 0 or s j minus s y i plus delta and this should be I have to take sum of this for all j which is not equal to y i. So, here you find that what I get is if s j minus y i this particular value equal to 0, then delta which is a positive constant is greater than 0 so, output will be 0.

Whereas, if s j minus y i this is minus delta; that means, y i minus s j is equal to plus delta then s j minus y i plus delta will be equal to 0, if you take the max function the output here also be 0 and it will remain 0 as long as s y i minus s j is greater than delta that is s j minus y i is less than minus delta. So, for all those cases this max function will be giving an output 0 and the loss component the contribution to this loss function L i in that case will also be equal to 0 and L i will be positive as long as the margin is less than delta.

So, this particular loss function confirms ensures that score for category omega I will always be greater than the score for any other category s j at least by a margin delta right.

(Refer Slide Time: 28:05)

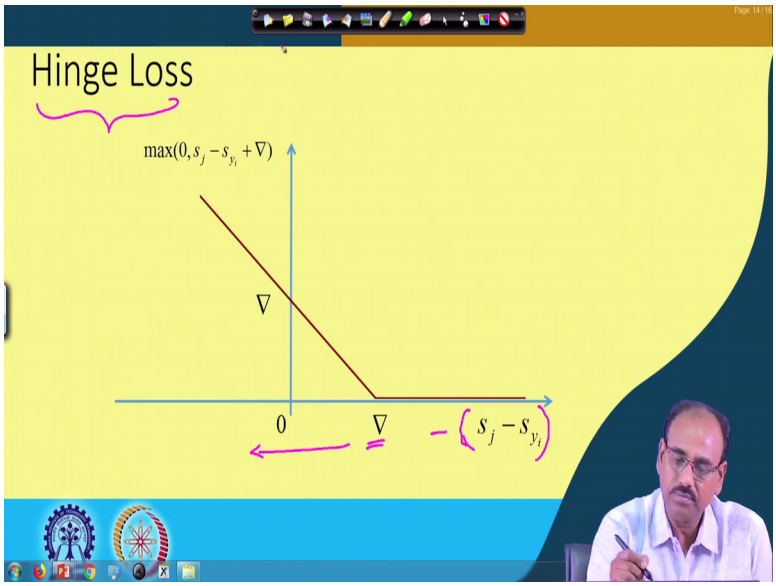


So, if I this is an example that suppose I have taken an y i which belongs to class II and the score function is given by 10, 30 minus 20 minus 20, 25 and if you compute the loss function then you find that for the first one the loss function will be equal to 0, because 30 is greater than 10 by a factor by a margin which we had in this case we have assumed to be 10 by it is; 30 is greater than 10 by a margin which is more than 10.

Coming to this one again 30 is more than minus 20 which is score for the third category by more than 10 come to the third category fourth category for which the score is 25 now we find that the margin for category 2 to which my input vector belongs so, this margin from category 2 the difference between the score for category 2 and category 4 is only 5 which is not sufficient because I have assumed margin to be 10.

So, this fourth component that actually gives me a nonzero loss function over here and that nonzero loss function is equal to 15; whereas, the others gives me a loss value which is equal to 0, that gives me the overall loss function a L i which is equal to 15 ok.

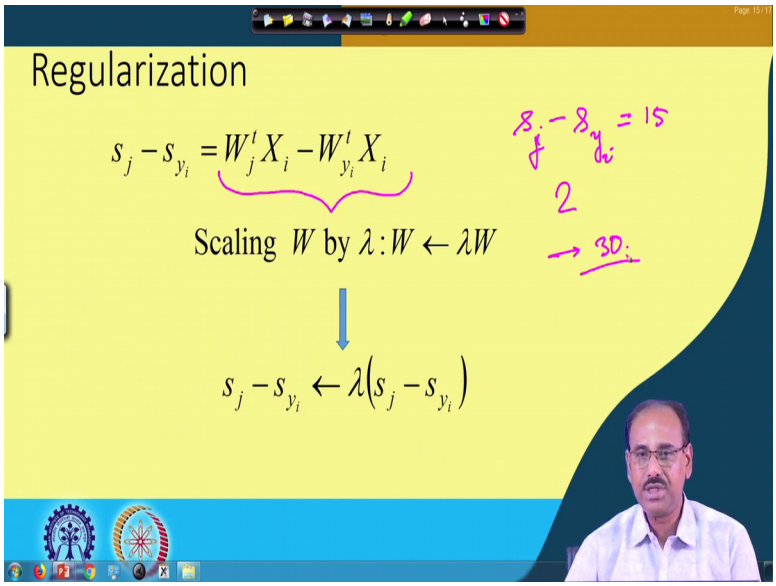
(Refer Slide Time: 29:35)



And, if you plot this loss function I plot this loss function which is max of 0 s j minus s y i plus delta versus s j minus s y i, you find that as long as s j minus s y i sorry it should be just negative of this s y i minus s j if I take the negative value of this; that means, s y i minus s j as long as it is greater than delta then your loss function is 0. The moment it becomes less than delta loss function goes on increasing which is in this direction.

And, because of nature of this variation of loss function this is what is known as hinge loss. So, in case of linear machine which minimizes hinge loss, you find that we are talking about a margin so, the score between the correct class and in incorrect class the difference of these scores must be greater than margin delta which is similar to the two class of vector machine that we discussed earlier this is what is known as multi class support vector machine. So, in case of multiple class support vector machine it tries to minimize the hinge loss as shown in this case.

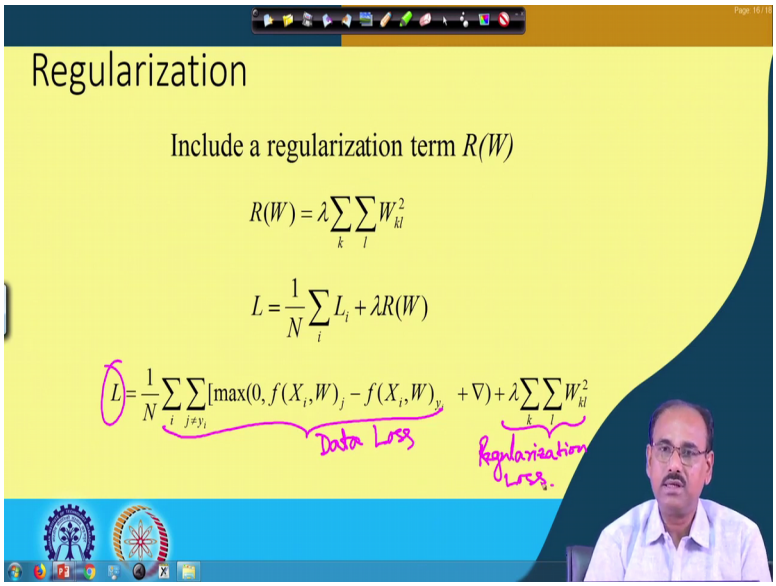
(Refer Slide Time: 31:19)



Now, this hinge loss is not sufficient. The reason being, if I multiply my weight matrix W by a constant say lambda which is greater than one you find that the difference between s j and s y i is nothing but if I take the difference of the dot products of the j th row of weight matrix W and y i th row of j th matrix w. So, s j minus s y i is nothing, but the difference of these two dot products.

So, accordingly if I scale of W by lambda these differences is also going to be scaled up. So, if for some W my s j minus s y i was say 15, now if I scale of W by 2 then this s j minus s y i will be 30. So, for different values of W I will have different differences of the score function. So, I have to choose that which value has to be proper among all these different possibilities of W, I have to choose a proper W; that means, I have to have impose a condition on W itself and that is what is known as regularization.

(Refer Slide Time: 32:49)



So, when you go for regularization you include a regularization term in your loss function. So, this regularization term is what is R W and usually the regularization term which is included is the L 2 norm of this weight matrix. So, accordingly you modify your cost function as L is equal to this was the function corresponding to your hinge loss and you find that the hinge loss depends upon the data along with your parameters of the linear machine.

So, this is a component which is called data loss and I also impose a condition a regularization term which is L 2 norm of my weight matrix which is delta W delta times sum of W kl; W kl is the kl th element of the weight matrix squared take the sum of this over l and k and this is what is known as a regularization loss.

So, now overall loss function includes two terms; one is the data loss and other one is the regularization loss. So, while designing this multi class support vector machine I have to optimize or we have to minimize this overall loss function. So, with this I will stop today.

Thank you.