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

Lecture - 13
Linear Machine

Hello welcome to the NPTEL online course on Deep Learning.

(Refer Slide Time: 00:31)

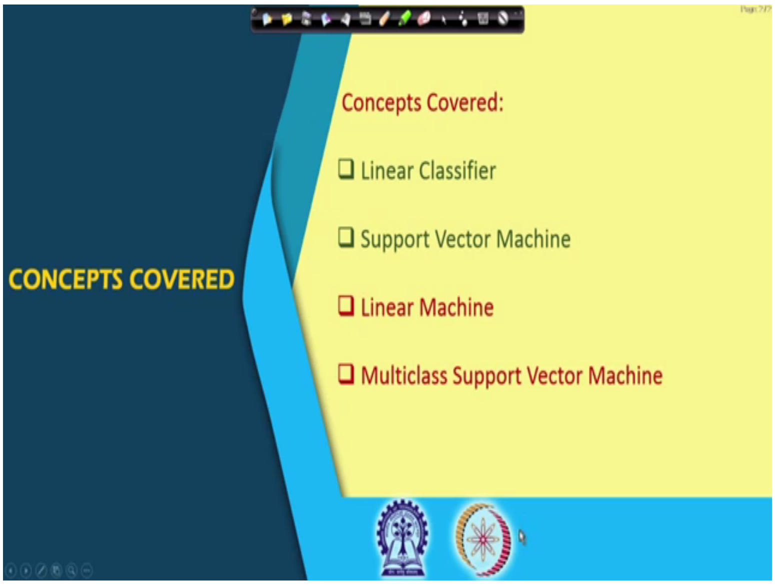
NPTEL ONLINE CERTIFICATION COURSES

Course Name: Deep Learning
Faculty Name: Prof. P. K. Biswas
Department : E & ECE, IIT Kharagpur

Topic
Lecture 13: Linear Machine

You remember in the previous class we have talked about the linear classifier and the support vector machine. So, what we have done in case of a linear classifier is that we assumed that we have a number of training samples.

(Refer Slide Time: 00:48)

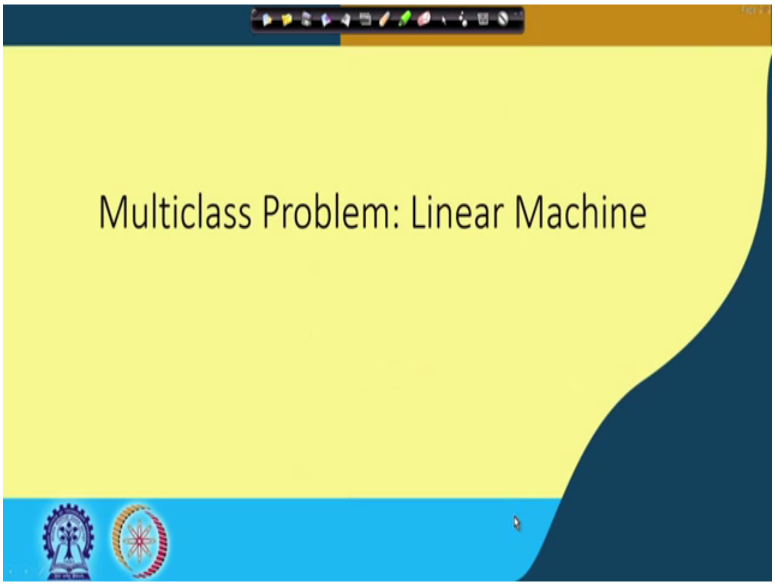


CONCEPTS COVERED

Concepts Covered:

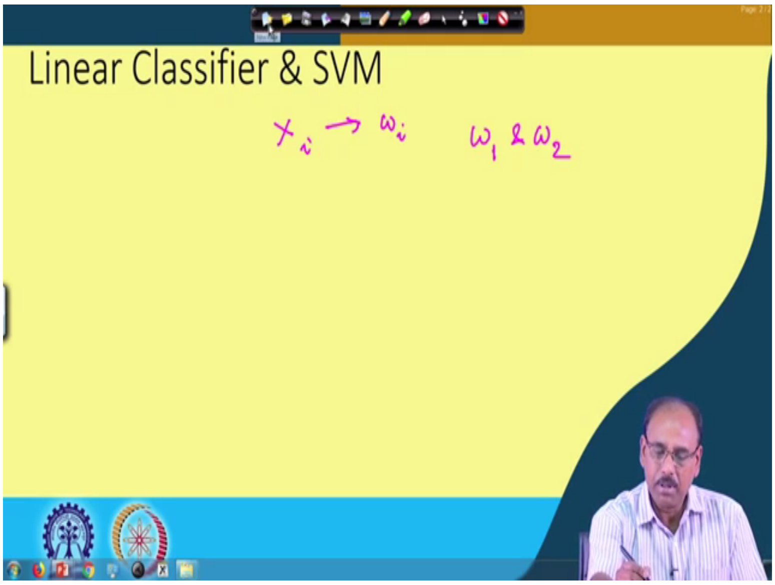
- □ Linear Classifier
- □ Support Vector Machine
- □ Linear Machine
- □ Multiclass Support Vector Machine

(Refer Slide Time: 00:48)



Multiclass Problem: Linear Machine

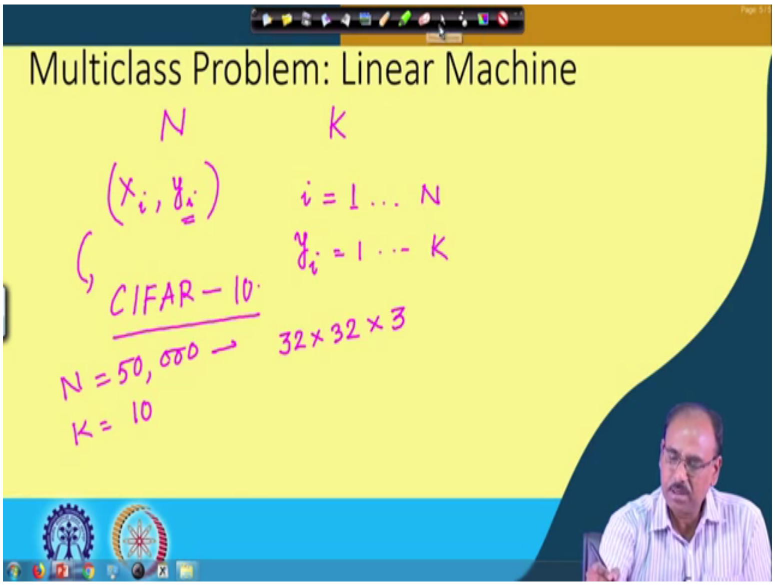
(Refer Slide Time: 00:54)



Linear Classifier & SVM

$X_i \rightarrow \omega_i$ ω_1 & ω_2

(Refer Slide Time: 00:56)



Multiclass Problem: Linear Machine

N K

(X_i, y_i) $i = 1 \dots N$

$y_i = 1 \dots K$

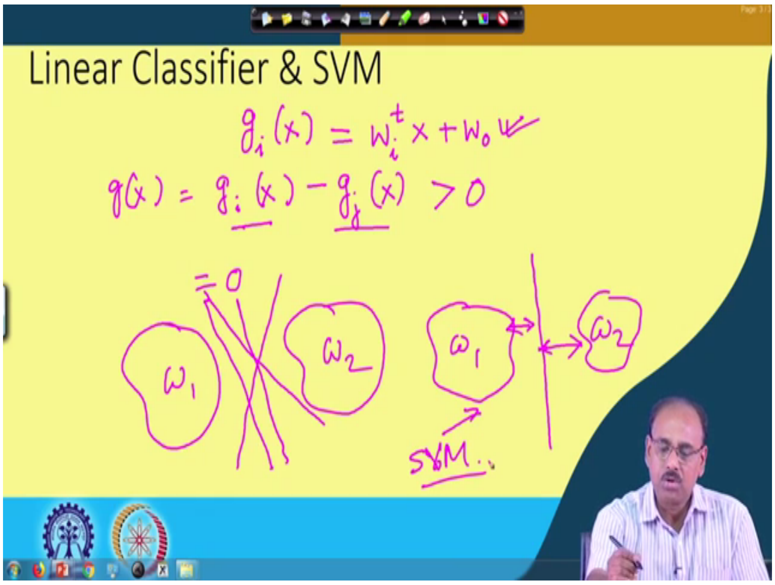
CIFAR – 10

$N = 50,000 \rightarrow$ $32 \times 32 \times 3$

$K = 10$

Say training samples y i which were assumed to belong to class omega i, in the sense that I had a number of training samples from 2 different classes, the class omega 1 and class 2 and using this training samples, we wanted to find out a separating plane which separates the samples belonging to class omega 1 and class omega 2.

(Refer Slide Time: 01:35)



Linear Classifier & SVM

$g_i(x) = w_i^t x + w_0$

$g(x) = \underline{g_i(x)} - \underline{g_j(x)} > 0$

Now before that if you remember when he talked about that discriminant function; for every class we defined a discriminant function which was given by g i X and in case that the covariance matrix of the training samples coming from all different classes are same this g i X came out to be a linear one which is of the form w transpose x plus w naught. Of course, if the covariance matrices of the training samples coming from the different classes; they are not same, then we have seen that this g i X or the discriminant function of the different classes that does not become linear anymore, but it becomes a quadratic function.

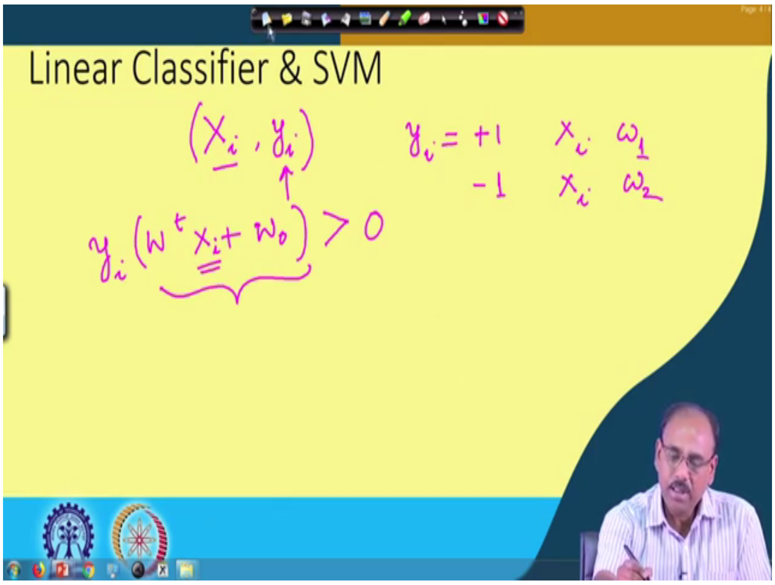
So, in case of linear classifier what we tried to find out is we tried to find out the boundary between 2 different classes and when the discriminant functions are linear then the boundaries between 2 different classes is also linear. Or in the other sense, the boundary between the 2 different classes is given by g X is equal to g i X minus g j X and if this becomes greater than 0. That means, for any given sample X if the discriminant function for class omega i is greater than the discriminant function for class omega j, in that case g i X minus g j X becomes greater than 0 and our conclusion is that X belongs to class omega i. Whereas, if g j X is less than g i X then; obviously, X belongs to class omega j and the boundary between the 2 classes is given by g X is equal to 0 when g i X and g j X are both of them are same.

So, in such case if this g i X or g j X they are linear as given in this case then g X also becomes linear which is the separating boundary between two different classes. So, given that if you are given a set of samples say belonging to class omega 1 and a set of samples belonging to class omega 2, then the boundary between these 2 classes omega 1 and omega 2 becomes a linear 1. So, when we talked about the linear classifier we assumed that the boundary is linear; that means the training samples or the samples belonging to class omega 1 and omega 2; they are linearly separable. And while designing linear classifier we did not really think of what is the distribution of the vectors belonging to class omega 1 and class omega 2 we simply assumed that they are linearly separable.

And we have seen while designing linear classifier is that as over here you find that this straight line is not unique I can have line over here, I can have a line here, I can have a line here and so on. So, there are multiple number of solutions possible and out of this multiple number of separating planes some of the separating planes gives you lesser margin, some of the separating planes gives you higher margin.

So, there we had gone for support vector machines which ensures that the margin that you get is maximum; that means, your separating plane given say a set of samples to omega 1 and another set of samples to omega 2. It gives you a maximum margin from the samples belonging to class omega 1 and the samples belonging to class omega 2 and this is what was the M of support vector machine or SVM.

(Refer Slide Time: 05:44)



Linear Classifier & SVM

$$(\underline{X_i}, y_i) \qquad y_i = +1 \quad X_i \quad \omega_1$$

$$-1 \quad X_i \quad \omega_2$$

$$y_i \underbrace{(w^t \underline{\underline{x_i}} + w_0)} > 0$$

So, for all these cases what we have seen is that you are given a set of training samples for designing the linear classifier or for designing the support vector machine and those training samples are actually leveled; that means, for each sample I know that from which class or from which category that sample has been taken. Or in other words, every sample or feature vector X i comes with a level y i so, your samples are given in the form X i y i.

And in case of a linear machine or support vector machine in case of a linear classifier or a support vector machine that we have discussed so far, because we were concentrating on only 2 class problem. So, this y i which actually tells you from which category, this sample X i has been taken this y i can have 1 of the 2 values. So, we assumed that y i was is plus 1, if X i is taken from class omega 1 and y i is minus 1 if X i is taken from class omega 2.

And accordingly, we could have a single classification rule or unified classification rule which we said that w transpose X plus X w naught of this form. So, this vector X is X i this whole thing multiplied by y i that should be greater than 0, if X i is correctly classified by the weight vector w and the margin w naught. This is obvious because if it is if the sample X i is taken from class omega 1 then my condition is w transpose X i plus w naught should be greater than 0 for correct classification, if it is taken from omega 2 as the sample as omega 2 is on the negative side of the separating plane. So, I had to have w transpose X i plus w naught less than 0 and for samples taken from class omega 2 because y is minus 1.

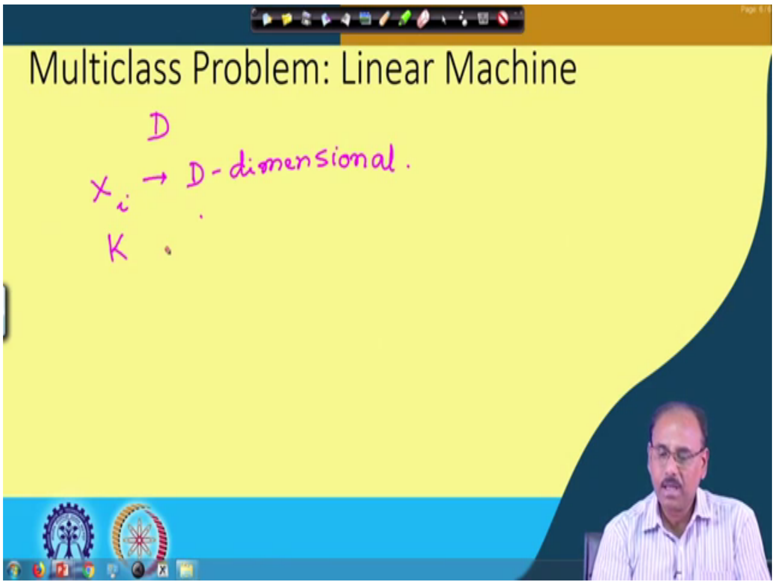
So, if I multiply this expression w transpose X i plus w naught by y i which is minus 1 in this case that will be greater than 0 so, I get a uniform classification rule. So, while designing the support vector machine or while designing the linear classifier we have taken advantage of this. So, for any X i when y i w transpose X i plus w naught was less than 0 we had taken that that particular w and w naught could not classify X i properly. So, the vectors have to be modified and that is how we have designed the linear classifier and also the support vector machine.

So, in this case also we assume that we are given set of feature vectors for training purpose. So, in a multi class problem now w i will not take only 2 values now because now with a multiple number of classes. So, what we assume is that, suppose I have got N number of feature vectors and I have say k number of categories or k number of classes. So, a training vector which is given in the form X i y i as I have got N number of feature vectors given for training. So, this i will vary from 1 to N as there are N number of feature vectors and my categories there are k number of categories and this y i is an index to the category. So, this y i will vary from 1 to k so, y i is an index to the category say for example, if I have 5 categories corresponding. So, corresponding to say apple, bird, then cat, dog, car and so on, if apple is the first category for apple y i will be is equal to 1.

If cat is the third category then for all the vectors belonging to cat class will have y i value is equal to 3 and so on so, for example, there is a public database called CIFAR 10. So, this CIFAR 10 database it contains 50,000 images so, N is equal to 50,000 and there are 10 categories. So, k is equal to 10 and the categories of images are images belonging to car images belonging to cat, images belonging to dog, images of sheep, images of horse and so on.

So, they are 10 such categories and there are 50,000 images and each of this image is of size 32 by 32 pixels and these are color images; that means, there are 3 planes red, green and blue so, number of pixels is 32 by 32 by 3.

(Refer Slide Time: 11:34)



Multiclass Problem: Linear Machine

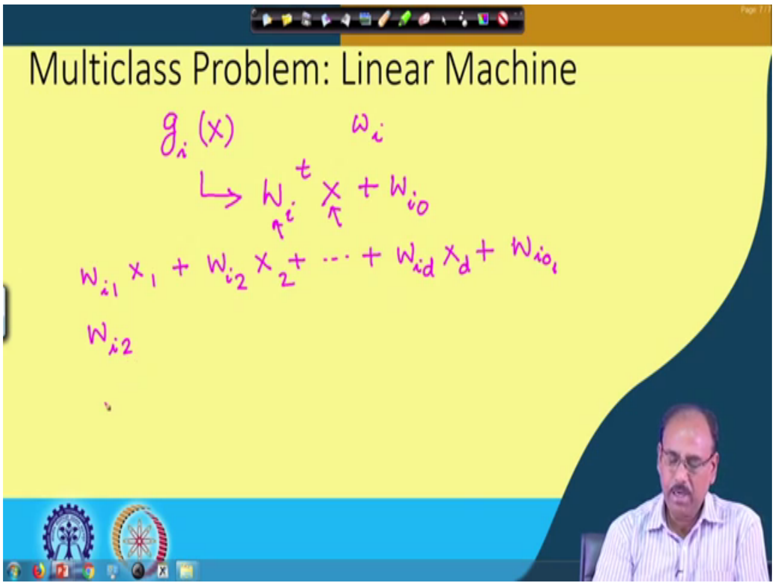
D

$X_i \rightarrow$ D-dimensional.

K

So, for discussion we will assume that every vector or feature vector given for classification purpose is of dimension D. So, our X i the feature vectors will be D dimensional and the number of categories k will be and there are k number of categories. So, I have D dimensional feature vectors and I have k number of categories.

(Refer Slide Time: 12:11)



Multiclass Problem: Linear Machine

$g_i(X)$ ω_i

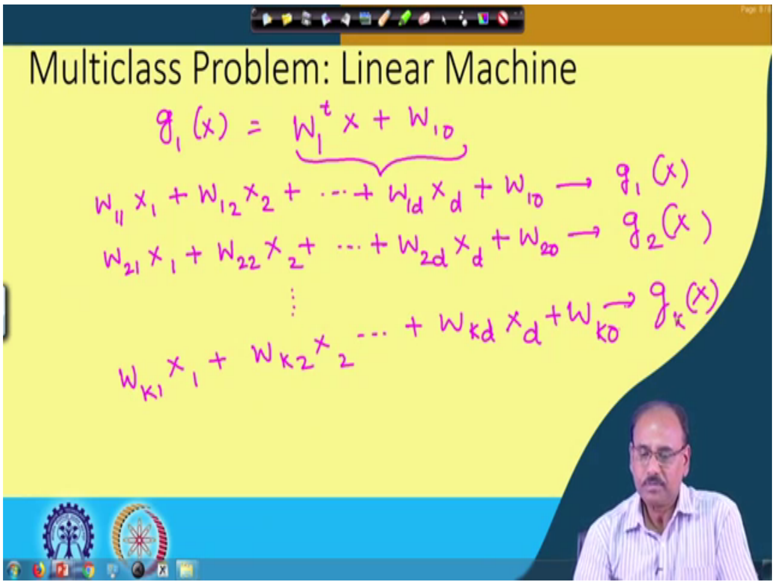
$\hookrightarrow W_i^t X + W_{i0}$

$W_{i1}X_1 + W_{i2}X_2 + \cdots + W_{id}X_d + W_{i0_1}$

W_{i2}

So, using this now you find that if I go for the same discriminant function we had the discriminant function of the form g i X for category omega i or ith category and this g i X as we are assuming that our discriminant functions are linear discriminante functions. So, this g i X is nothing, but of the form w i transpose X plus w i naught, where this X is a D dimensional vector, w i is also a D dimensional vector and you find that if I expand this expression then my g i X becomes of the form w i1 X 1 plus w i2 X 2 plus as there are d number of components. So, I will have w id X d plus I can write this as w i0 1 which is the bias term so, this is what is my g 1 X for category omega 1. Similarly, for category 2 g 2 X will be w aa sorry I am taking i is equal to 1. So, this expression will be different let me put it like this.

(Refer Slide Time: 13:53)



Multiclass Problem: Linear Machine

$$g_1(x) = \underbrace{W_1^t x + W_{10}}$$

$$W_{11}x_1 + W_{12}x_2 + \cdots + W_{1d}x_d + W_{10} \rightarrow g_1(x)$$

$$W_{21}x_1 + W_{22}x_2 + \cdots + W_{2d}x_d + W_{20} \rightarrow g_2(x)$$

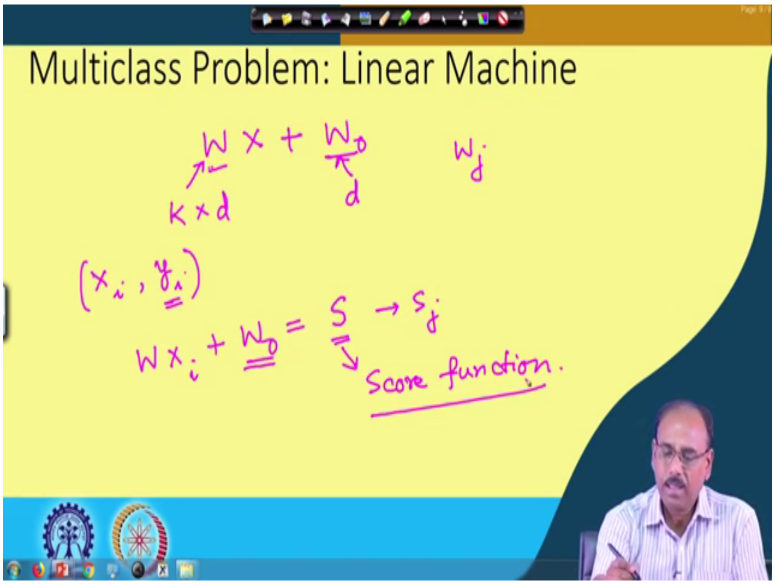
$$\vdots$$

$$W_{K1}x_1 + W_{K2}x_2 \cdots + W_{Kd}x_d + W_{K0} \rightarrow g_K(x)$$

So, I want to find out g 1 X which is nothing, but w 1 transpose X plus w 1 naught, as w 1 has got d number of components X also has got d number of components. So, I can expand this expression in the form w 11 X 1 plus w 12 X 2 plus w 1d X d plus w 10, this is what is my g 1 x. Similarly g 2 X becomes w 21 X 1 plus w 22 X 2 plus continue like this w 2d X d plus w 2 naught which is the bias for category 2.

And as have got K number of classes I will have K number of discriminate functions. So, g k X will be of the form w k1 X 1 plus w k2 X 2 w kd X d plus w d w k0. So, this is your disconnect function g 1 x, this is the discriminant function g 2 x, this is the discriminant function g k x. Now given this k number of linear equations you find that I can represent this k number of linear equations in the form of matrix equation.

(Refer Slide Time: 16:01)



Multiclass Problem: Linear Machine

$$\underset{k \times d}{W} X + \underset{d}{W_0} \qquad W_j$$

$$(X_i, y_i)$$

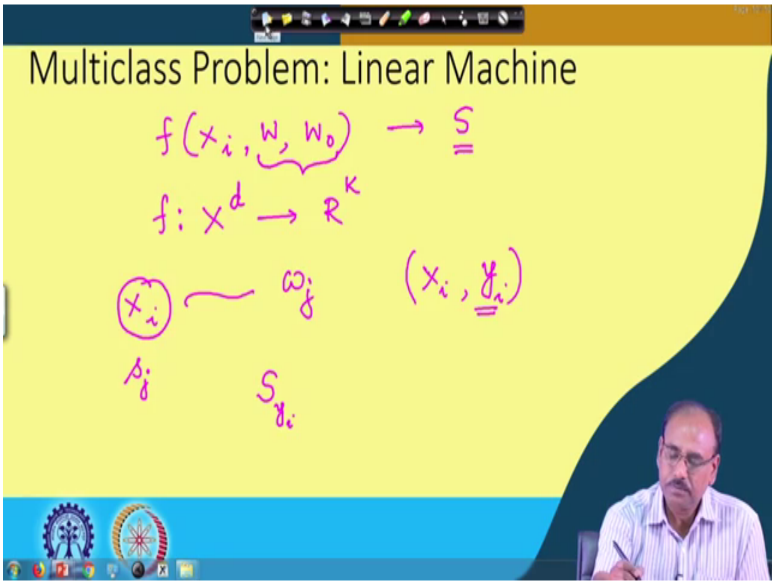
$$W X_i + W_0 = S \to S_j$$

Score function.

So, the matrix equation simply becomes w X plus w naught, where this w is a matrix having k number of rows and d number of columns. So, this w is a k by d matrix and the w naught is a bias which is a column vector having d number of components. So, a linear machine is a one which is specified by this matrix w and this bias vector which is w naught so, they are the parameters of a linear machine. So, once I have this for given any unknown vector X i or if I take the training vector X i which belongs to omega i sorry which belongs to omega I so, corresponding that my index is y i. So, for this X i y i pair if I compute this expression w X i plus w 0, this will give me a column vector let me call that column vector to be s.

So, what this column vector gives, you find that here what you are computing is you are multiplying X i with every row of w. So, if I take the j th row of w to be w j. So, this X i is you take the dot product of X i with the j th vector in this matrix w and multiply and add to that the j th component of your bias vector w naught to give the j th component of s or s j so, we call this s to be a score function. So, what this linear machine is doing.

(Refer Slide Time: 18:20)



Multiclass Problem: Linear Machine

$$f(x_i, \underbrace{W, W_0}) \rightarrow \underline{\underline{S}}$$

$$f: X^d \rightarrow R^K$$

$$\boxed{x_i} \;—\; \omega_j \qquad (x_i, \underline{\underline{y_i}})$$

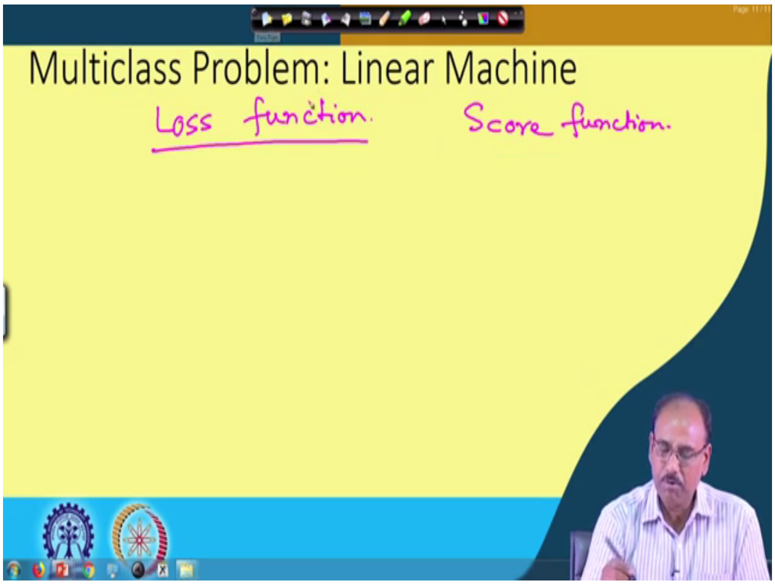
$$s_j \qquad S_{y_i}$$

The linear machine is nothing, but a function f which operates on an input vector i the parameters of the linear machine are say, w and w naught and this gives you a score function which is s again a D dimensional vector. So, this operation I can say that it is a mapping which mappings map my D dimensional feature vector X d into a score which is of dimension k and every component of this core vector of this score gives you the score for the corresponding class.

So, given my input vector X i the score component s j tells me that what is the score of this input vector X i to a category omega j as given by the linear machine decided by which are having the parameters w and w naught so, this is what the linear machine does. And now if my training vector says that this X i is taken from class omega i or the corresponding class index is y i then; obviously, what I would like to see is that for the score function s the y i th component of this core function should be maximum. So, that the classification of the categorization as given by this linear machine is correct.

And if it is not correct then I have to go for correction of the parameters or updation of the parameters of the linear machine. So, that the output classification becomes or the error in the categorization or the classification is minimized.

(Refer Slide Time: 20:45)



Multiclass Problem: Linear Machine

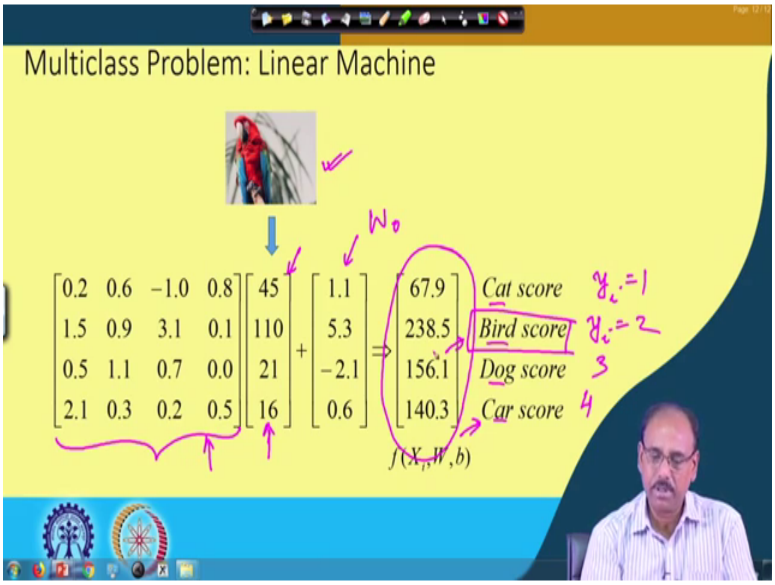
Loss function.

Score function.

So, what I want to do is, I want to now define something called a loss function. So, one thing we have seen is the score function the linear machine gives you the score function and the score function indicates that for a given feature vector what is the core for what is the score for different process or for different categories. And I want that this score should be similar to the ground truth it should be same as the ground truth; that means, if my feature victor belong to class omega 1 or it will it is from category 1. In that case, in this core vector that I get the first component should be maximum and all other components should be less than the first component. So, that my classification is correct.

If it is not, then I had to go for updation of the parameters of the linear machine and this parameters are nothing, but w and w naught and by updation I should be able to tune the parameters so, that I get a correct classification ok.

(Refer Slide Time: 22:04)



Multiclass Problem: Linear Machine

$$\underbrace{\begin{bmatrix}0.2&0.6&-1.0&0.8\\1.5&0.9&3.1&0.1\\0.5&1.1&0.7&0.0\\2.1&0.3&0.2&0.5\end{bmatrix}}_{W}\begin{bmatrix}45\\110\\21\\16\end{bmatrix}+\overset{W_0}{\begin{bmatrix}1.1\\5.3\\-2.1\\0.6\end{bmatrix}}\Rightarrow\begin{array}{c}\left[\begin{array}{c}67.9\\238.5\\156.1\\140.3\end{array}\right]\\ f(X_i,W,b)\end{array}\begin{array}{ll}\text{Cat score} & y_i=1\\ \boxed{\text{Bird score}} & y_i=2\\ \text{Dog score} & 3\\ \text{Car score} & 4\end{array}$$

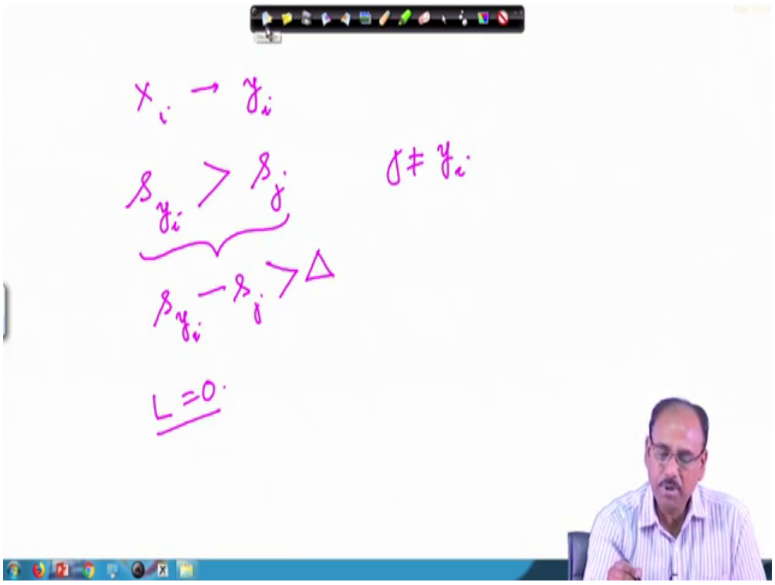
So, what I want is ok, before I define this loss function I want to show this with an example say for example, over here what we are doing is, we have taken an image and this image is the is flattened out to convert this to a vector. So, how do you flattened it out? You can take every column of the image and concatenate the columns to get a vector right.

So, if I assume that this image is converted to a vector having 4 components and this is my linear machine, the linear machine takes care of 4 different categories the categories are cat, bird, dog and car. So, in this case cat is the first category, bird is the second category, dog is the third category and car is the fourth category. So, for cat y i will be equal to 1 for bird, y i will be taken as 2 for dog, it will be taken as 3 and for car it will be taken as 4 which are the indices to all these different categories.

So, you find that when you multiply this vector with this matrix w and this is my bias vector w naught I get this output which is the score and over here because this is an image of a bird you find that the bird score is maximum. If these parameters are not properly tuned then it is possible that the score for car will be might be more than the score of bird, in which case my classification that I get the linear machine the classification it gives that is an erroneous classification and I should be able to modify the parameters or update the parameters. So, in order to do this what I go for is, I define a loss function I will come to this a bit later.

So, what is the loss function?

(Refer Slide Time: 24:22)



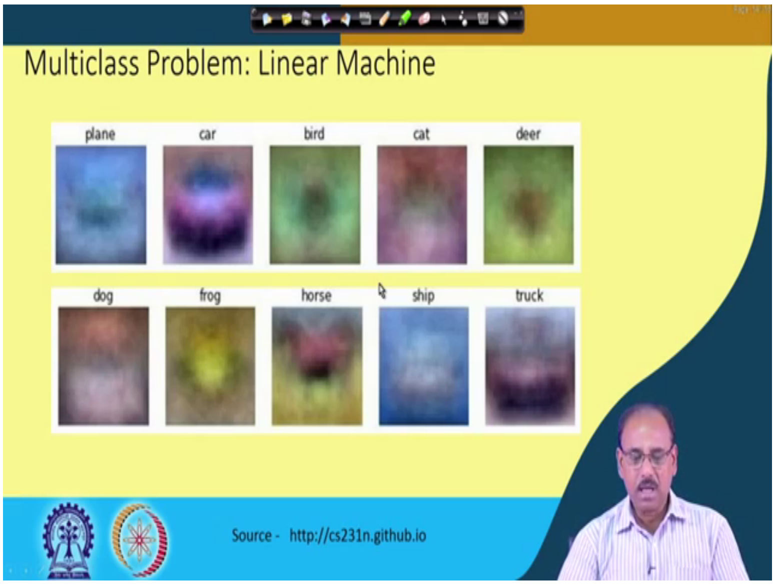
As I said that it my classification is not correct or even if my classification is correct, but I want to have better confidence over the classification or in other words I want that if the vector X i actually belongs to category omega i or the index is y i then the in the score function s y i should be greater than s j for all j not equal to y i this is what I want and just that s i is more than s j I may not be satisfied with this, because I want to have more confidence over this classification. So, for that I may like to have s y i minus s j should be greater than some delta.

So, that is your confidence factor ; that means, it is not only that s i s y i should be more than s j, but s y i should be more than s j at least by a factor of delta. And only then I will be satisfied and in such case I will assume that my loss function will be equal to 0.

In other cases I will assume that my loss function will not be 0, but I will have a finite loss function and that loss value I want to minimize to get the correct classification. I will come to this loss function in details in our previous class, but before that let me just tell you what is the interpretation of this linear machine. So, as we have seen so far that in this linear machine what we are doing is, for every class or for every category I have a vector which is a row in the vector w and for this input vector I am taking the dot product of this input vector with the i th row of my matrix w the parameter matrix w and this dot product gives me the score along with the bias term the score for the i th class of the given vector i. And as you know that when you take the dot product of 2 different vectors the dot product gives you a measure of similarity, that is if the 2 vectors are similar then the dot product will be higher, if the 2 vectors and dissimilar then the dot product will be lower.

So, going by that I can consider this linear machine the operation given by a linear machine is something like a template matching operation or in other words every row in the parameter vector in the parameter matrix w is a template of that corresponding class and you are trying to find out you are trying to make match your input vector with that template and in fact, in CIFAR 10 database when you train the linear classifiers.

(Refer Slide Time: 27:50)



Multiclass Problem: Linear Machine

Source - http://cs231n.github.io

You can see that these are the different classes different categories which the CIFAR 10 database contains you have the category of plane, car, bird, cat, deer, dog, frog, horse, sheep and truck.

And once the linear machine is properly trained it is properly trained then all the vectors of your weight matrix w if you fold it back to form an image, then you will find that those vectors represent the templates of these different categories. Say for example, here this is the template of a plane, this is the template of a car, this is the template of horse, this is the template of truck and so on. So, these weight vectors that I get for different classes are nothing, but these templates and what the linear machine is doing is it is matching your input vector with these templates.

So, I will stop here today in your in our next lecture we will talk about the loss functions details of the loss functions and how these loss functions can be used for designing of the linear machine.

Thank you.