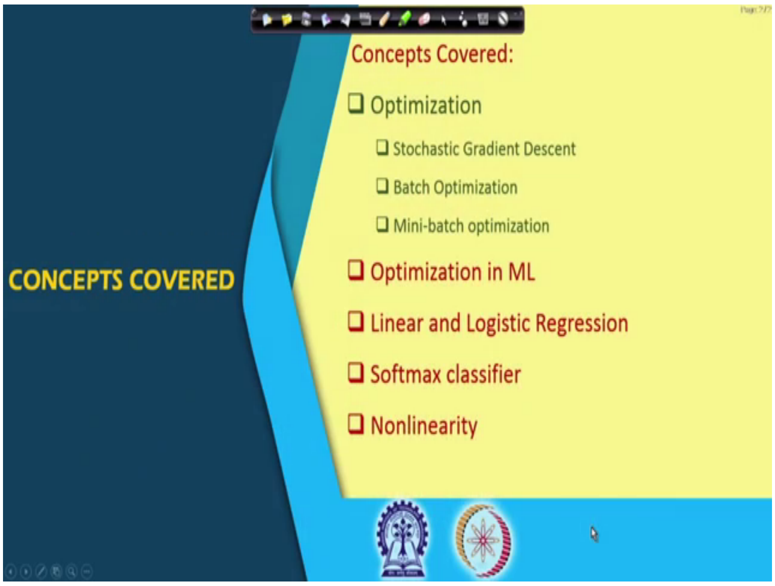
Deep Learning
Prof. Prabir Kumar Biswas
Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology, Kharagpur

Lecture - 17
Optimization Techniques in Machine Learning

Hello welcome to the NPTEL online certification course on Deep Learning.

(Refer Slide Time: 00:32)



CONCEPTS COVERED

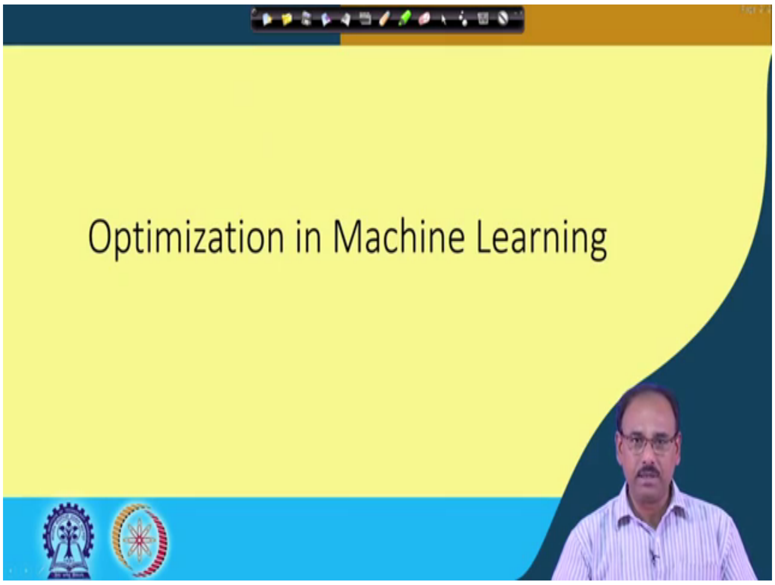
Concepts Covered:

- ❑ Optimization
  - ❑ Stochastic Gradient Descent
  - ❑ Batch Optimization
  - ❑ Mini-batch optimization
- ❑ Optimization in ML
- ❑ Linear and Logistic Regression
- ❑ Softmax classifier
- ❑ Nonlinearity

You remember in the previous class we have started our discussion on Optimization Techniques. And, we have talked about the different optimization techniques like, stochastic gradient descent, we have talked about batch optimization, and we have also talked about mini batch optimization.

In today's lecture we are going to talk about optimization in machine learning. In particular that how optimization in machine learning applications differs from general optimization problems. We will also talk about linear and logistic regression, sortmax classifier and we will also talk about the non-linearity, how non-linearity is important in case of deep learning or machine learning applications.

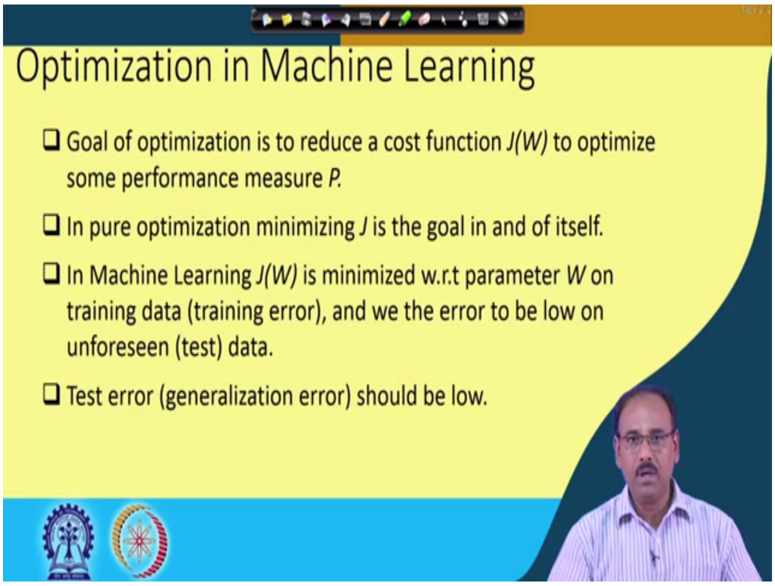
(Refer Slide Time: 01:27)



Optimization in Machine Learning

So, let us first talk about how machine learning, optimization in machine learning is different from general optimization techniques.

(Refer Slide Time: 01:39)



Optimization in Machine Learning

- ❑ Goal of optimization is to reduce a cost function *J(W)* to optimize some performance measure *P*.
- ❑ In pure optimization minimizing *J* is the goal in and of itself.
- ❑ In Machine Learning *J(W)* is minimized w.r.t parameter *W* on training data (training error), and we the error to be low on unforeseen (test) data.
- ❑ Test error (generalization error) should be low.

So, what you do in case of optimization? The goal of optimization is to reduce a cost function given by say J W where W is the parameters, which defines your machine learning algorithm. So, you optimize this cost function J W or minimize the cost function J W, in order to optimize some performance measure of your learning algorithm or machine learning model.

So, what is the performance measure in case of machine learning, that we have discussed. So, far the problems that we have considered are mainly categorization or classification problems, that is given objects or signals belonging to different categories, we try to identify that which signal or which object belongs to which kind of category.

So, if your categorization or classification is correct, there is no error if the classification is incorrect you incur some error. So, the performance measure with respect to machine learning is how accurate, your classification or categorization problem is decision is. And, in order to do that you minimize a cost function J W, which is this cost function J W is comprises of the loss that you incur during classification of the training data.

Now, if you talk about the pure optimization problem in case of pure optimization problem the optimization criteria the cost function J is minimized and that is the goal in and of itself. In other sense suppose you are given a set of observations in 2 dimension, that is I may have x y pairs, where I assume that X is an independent variable and y is a dependent variable. And, if there is sufficient reason to believe that the relation between X and Y is linear then given a set of observations what we would like to do is we would like to fit a minimum error straight line passing through the set of observed data.

And, that is the M and when you try to find out this straight line, which minimizes the error actually you go for minimization of sum of squared error. And, once the line is fit your goal is satisfied. But, when you talk about machine learning, in case of machine learning, we minimize the loss function or cost function J W, where W is the parameter of your machine learning algorithm or the machine learning model.

But, this minimization is done on the training data or the training samples or in other words what you try to minimize is the training error. But, what is our actual aim our actual aim is not to classify the training data, because for the training data the classes are already known. So, what is the great thing about classifying the data, which are known already?

So, our aim is not tooked us correctly classify the training data, but our aim is that the machine that you have trained using the training data or the level data, that same algorithm or the machine has to be used for classification of unknown or unforeseen data, which the machine has not seen before. And for this unknown data the class belongingness is not really known, because if it is known then I do not have to classify.

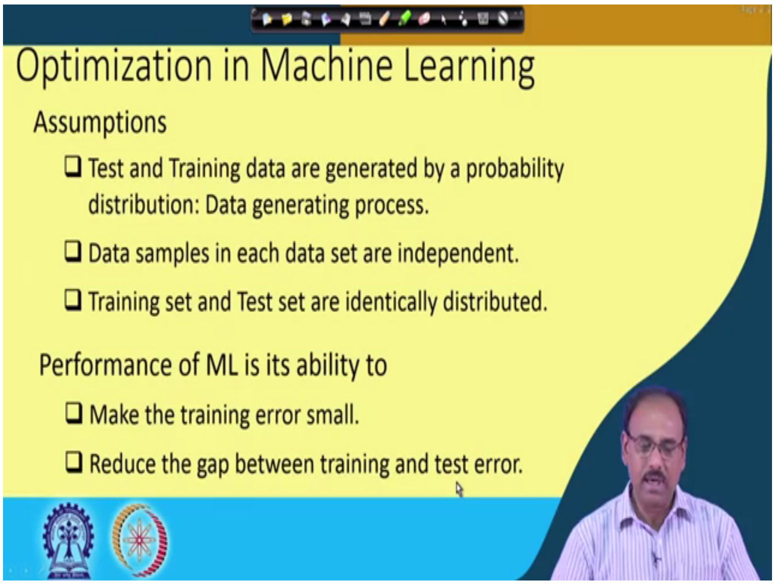
So, for this what my aim is that though I am training the machine or minimizing the error on the training data, but my aim is that the same machine has to perform well on the actual data or the test data. So, I want that not only the training error to be minimized, but also the test error is to be minimized.

So, how do you know the test error, because as I said that for your real life classification problem that class belongingness of the data is not known. So, what you do is you set aside a set of level data, which unknown as test data, for which the classes are known, but you do not use that set of data or test set for training the machine. So, in that case I can compute what is the error given by the machine while testing, but while training you use one set of data which is training set for testing you set another set of data which is test set.

But, again when you come to actual application, when you want to deploy this machine learning model machine learning algorithm for classification or recognition, there the class belongingness of the data is not really known. So, accordingly you incorporate some test error, when you are testing the machine on the test set of data which are not actually used for training purpose. And, while doing so, you incorporate some error or some laws, which are the test which is known as the test error or generalization error.

So, in case of machine learning algorithms or optimization in case of machine learning, though I am optimizing the machine on the training data, but my aim is that the same machine should perform well on the test data as well as on the real life data.

(Refer Slide Time: 07:35)



Optimization in Machine Learning

Assumptions

- ❑ Test and Training data are generated by a probability distribution: Data generating process.
- ❑ Data samples in each data set are independent.
- ❑ Training set and Test set are identically distributed.

Performance of ML is its ability to

- ❑ Make the training error small.
- ❑ Reduce the gap between training and test error.

So, that is the basic difference between a machine learning algorithm or optimization in case of machine learning algorithm.

So, and naturally because the test data has not been used for training the machine, how do we really know or how do we really guarantee, that the machine or algorithm which has been trained on the training data, will also perform well or give lesser tests error while testing on the test data or it will also give minimum error, when you actually deploy the machine for real life classification problem.

So, truly speaking this cannot be guaranteed, but under certain assumptions, we can say with certain degree of confidence, that the machine will also perform well on the test data. And, for that we make some assumptions popularly known as I ID assumptions. So, what you assume is that both of the training data and the test data, they are generated by a probability distribution, which is known as the data generation process or data generation model. We also assume that the data samples in each set in the data set are independent, that is one sample or one vector is independent of the other vector.

And, we also assume that the training set and test set are identically distributed. So, when we talk about the data distribution, the distribution of the training set of data and the distribution of the test set of data they are identical. And, that is what is known as I ID assumption that is independent identically distributed set of data. Normally, the performance of a machine learning algorithm is measured by it is ability to perform 2 tasks.

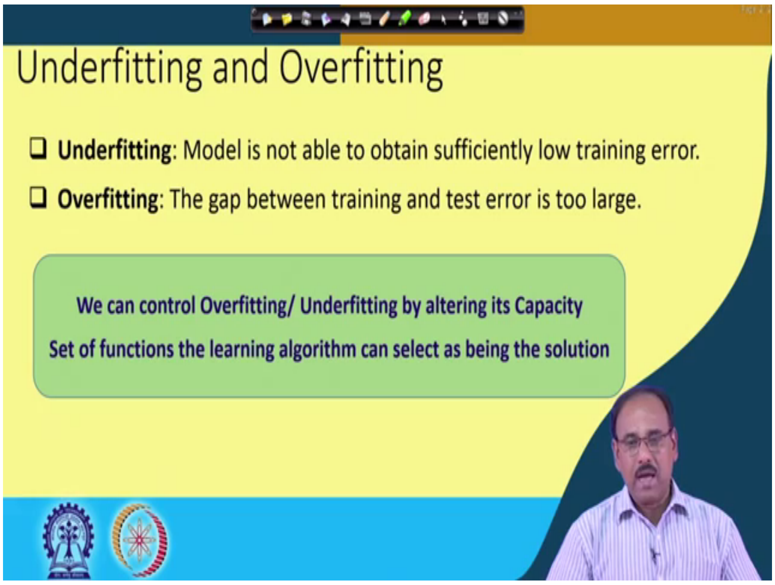
One of the tasks is that it has to make the training error small that is while training the machine on the training set of data, the training error that you incorporate that should be small and only when it is very small ideally it should be 0, we assume that the machine or your algorithm is properly trained, or it has learned their classes.

And, secondly, when you test it so, while testing on the test data and as we said before that the test data is not used during training the machine, though we know what are the class belongingness or the test data?. So, once the machine is trained on the training data, you use the same machine on the test data. And, in case of test data ideally we want that the error should be minimum or the error should be 0 in ideal cases, but practically that cannot be possible.

So, while applying the same machine on the test data we get some test error. So, again the performance of the machine learning algorithm is measured in terms of what is the gap between the between the training error and the test error. So, we always expect that whatever is the training error the test error should also be similar; that means the gap between the training error and the test error should be as small as possible.

So, these are certain assumptions and the under those assumptions we talked about how the machine learning algorithm actually performed.

(Refer Slide Time: 11:13)



Underfitting and Overfitting

- ❑ **Underfitting**: Model is not able to obtain sufficiently low training error.
- ❑ **Overfitting**: The gap between training and test error is too large.

We can control Overfitting/ Underfitting by altering its Capacity

Set of functions the learning algorithm can select as being the solution

Now, these assumptions and the performance measures that lead to 2 very important problems 2 challenging problems, one of the is known as under fitting problem and the other one is known as over fitting problem. So, what is under fitting problem? Under fitting problem is when your model or the machine is not able to obtain sufficiently low training error; that means, it is not performing well on the training data itself.

So, the training error is not acceptably low and as we said that in ideal cases, we expect this training error should be zero; that means, all the training samples which are given for training the machine they should be correctly classified and then only we get that to the training error to be 0. And, the over fitting says that the gap between training and test error is too large so, there are may be cases, that while training we have been able to design a machine or a model.

Where the training error is minimum but the same machine when it is applied on the test data, that test data is quite large; that means, there is a gap between the training error and the test error. So, over fitting says that your model has been able to capture even the minuet differences in the test data and while doing so, the model is so, tuned to that test data, that it is not it has lost the generalization and it is not being able to perform well on the test data. And, that is what is known as over fitting problem?

So, this problems of over fitting or under fitting can be controlled by altering another property of the model which is known as model capacity or capacity of the machine learning algorithm. And, this capacity is nothing, but a set of functions the learning algorithm can select as being the solution. So, we will just see after this and you remember that all the problems that we have considered we have discussed till now they are linear problems. We have assumed that given data belonging to two different classes, in the feature space, I can have a hyper plane in the feature space I can define a hyper plane in the future feature space.

Where all the training data belonging to one class falls on positive side of the hyper plane and all the training data, belonging to other class falls on the negative side of the hyper plane; that means, the data set is linearly separable, but what happens if the data set is not linearly separable. I may not be able to pass a plane or a hyper plane passing to the data sets belonging to two different classes. That means, in that case our linear assumption is not correct, is not sufficient, instead of a hyper plane, if I pass a curve or a surface a curved surface, then a curved surface may be able to separate those two a set of data belonging to two different classes.

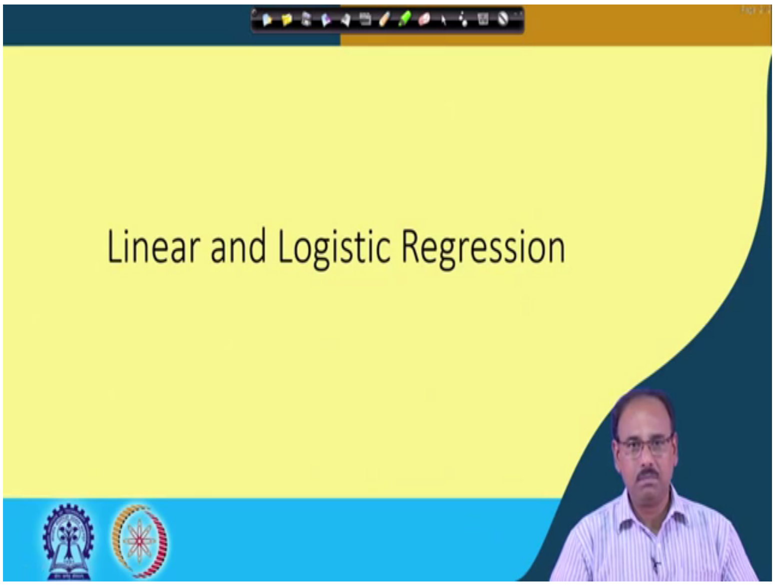
So, that is what is known as the state of functions. Always the linear function may not be sufficient I may have to go for functions of higher order may be quadratic may be cubic or even fourth order fifth order and so on, which is known as the capacity. So, if I control the capacity or the set of functions the training algorithm can adopt as required, then possibly we can have a control over the over fitting and under fitting problems.

And, there also it is not that that simple or that easy, because it is quite possible that say for example, data set which is sampled from a quadratic function. If, I want to fit a quadratic function, it will be properly the fit; that means all that data would be properly fit to the quadratic curve.

If I try to fit a linear surface then; obviously, I will incorporate some error, if I want to fit our data of the higher dimension say of a cup of higher dimension say order 5 or order 7 and so on. Then, it is possible that for the given set of data, I will have no loss there will be no error, but the data is too specific for that set of training samples. It is not general enough to perform well on unknown data set or test data set.

So, even selecting the capacity of the model or capacity of the learning measure of the machine learning algorithm is also a difficult task or a challenging task. So, here we have talked about that what is the difference between a general optimization task and an optimization task as cons as applicable to machine learning algorithms. Because, in case of machine learning algorithms though we are optimizing on the training set of data, but actually I want to perform on a different set of data which is test set of data right.

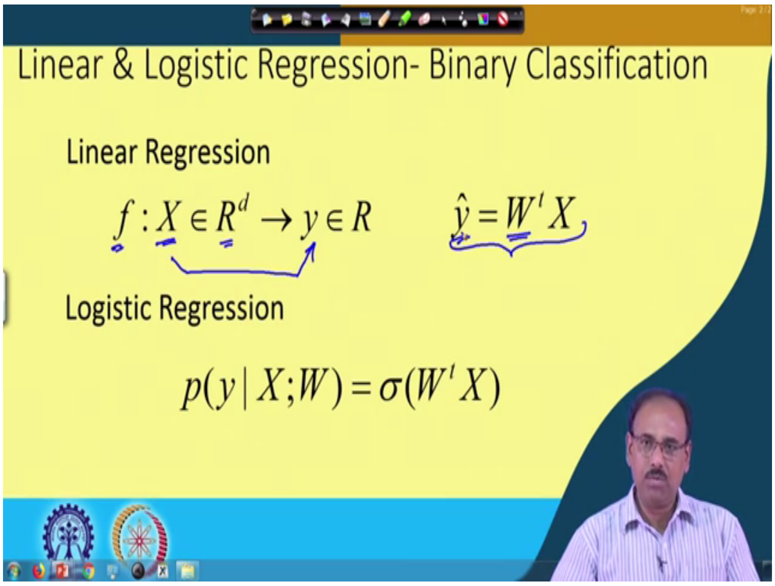
(Refer Slide Time: 16:41)



Linear and Logistic Regression

So, now I talk about to problems, which are known as linear regression and logistic regression and we will see the importance of this.

(Refer Slide Time: 16:55)



Linear & Logistic Regression- Binary Classification

Linear Regression

$$f : X \in R^d \rightarrow y \in R \qquad \underbrace{\hat{y} = W'X}$$

Logistic Regression

$$p(y | X; W) = \sigma(W'X)$$

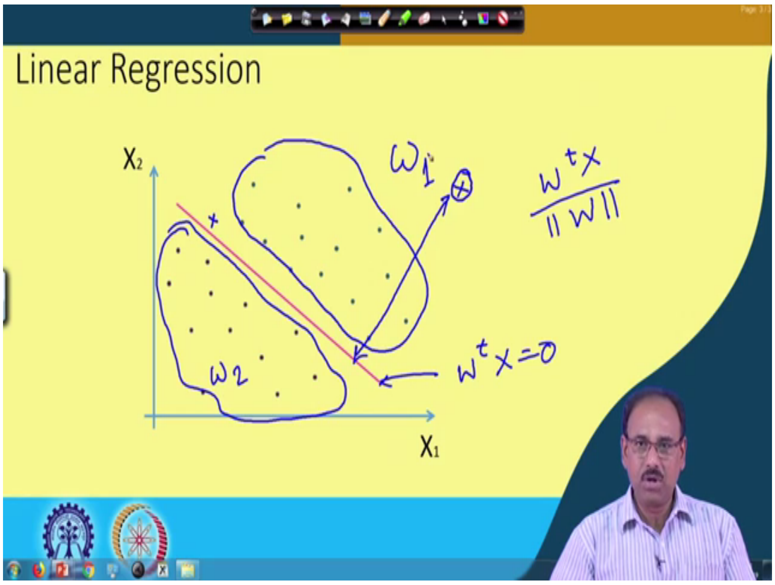
So, what is linear regression? You find that so far all the problems or the examples that we have considered as we have just mentioned now, that I have considered only the linear problems. That is given 2 sets of data I should be able to find out a straight line in 2 dimension or a surface or a plane in 3 dimension or the hyper plane in even higher dimension.

So, effectively there what I want to do is this model that maps a feature vector of dimension d to a scalar y which is a linear number. So, this is actually put in this form that this is a mapping, which maps a feature vector X of dimension d. So, we write it this way that X belongs to R d. So, it maps this feature vector X 2 and y or in the other words, that given feature vector X we want to predict what is y, what is the value of y?.

So, I can write this in the form of a linear equation of this form that y hat is equal to W transpose X, where W is the weight vector. And, based on the value so, while training what we try to do is we try to minimize the error between actual y, which is the 2 class that is given for the set of data and this y hat that is predicted. We want to minimize this error during the training operation. And, during classification what we do is based on this predicted value of y which is y hat, we take a decision that way there we should classify X 2 plus 1 or we should classify X 2 class 2.

So, for a binary classification problem that we have done previously we have discussed previously, we have seen that if y hat is positive; that means, if it is greater than 0, then we classify X 2 class 1 and if y hat is negative that it is less than 0, then we classify X belonging to belong to class 2.

(Refer Slide Time: 19:27)



Linear Regression

Now, what is this linear regression actually? If, you look at this expression that is W transpose X given this example here, we have taken the same example in your previous discussion, that we assume that this is the set of data, the set of feature vectors which belong to class say omega 1 and this is another set of feature vectors that belong to class omega 2.

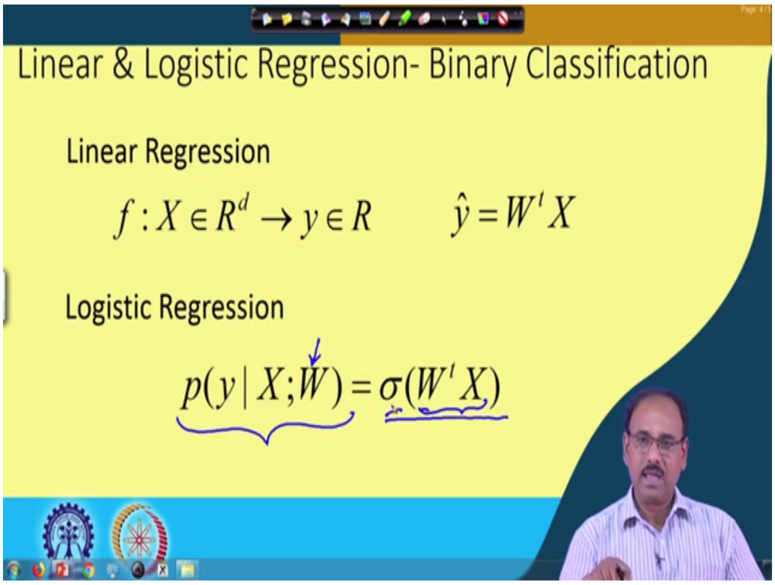
So, for every point and this is my hyper plane given by the equation W transpose X equal to 0. So, you find that for every point belonging to omega 1 your W transpose X will be greater than 0, for empty sample belonging to class omega 2 W some transpose X will be less than 0. And, in fact, this W transpose X upon mod of W that actually gives you an idea of what is the distance of perpendicular distance of the vector X from the plane W transpose X equal to 0.

So, that measure gives us a confidence of how accurate your classification is, because given these 2 sides if I have a vector over here, which is far away from the separating plane in such cases; obviously, I can tell with high degree of confidence that my classification result is correct. Whereas, if I have a data somewhere over here. For the distance of this data of this vector, from the separating plane is very small my confidence level is not that high. Though here the distance becomes some greater than 0 as W transpose X becomes greater than 0.

So, I classified this to belong to class omega 1 but my confidence level is not that high as I have confidence in this particular case. So, I have a physical interpretation of the value of W transpose X that it says, that how well inside the class the data is. And, if it is very high W transpose X is very high on the positive side, then with guarantee I can say that yes this belongs to class omega 1, I am and if it is very low that is W transpose X is high magnitude, but it is negative then I can tell with confidence that yes it belongs to class omega 2.

But, if the value is low my confidence level comes down. So, I can have another interpretation of this measure that is what is given by logistic regression.

(Refer Slide Time: 22:18)



Linear & Logistic Regression- Binary Classification

Linear Regression

$$f : X \in R^d \rightarrow y \in R \qquad \hat{y} = W^t X$$

Logistic Regression

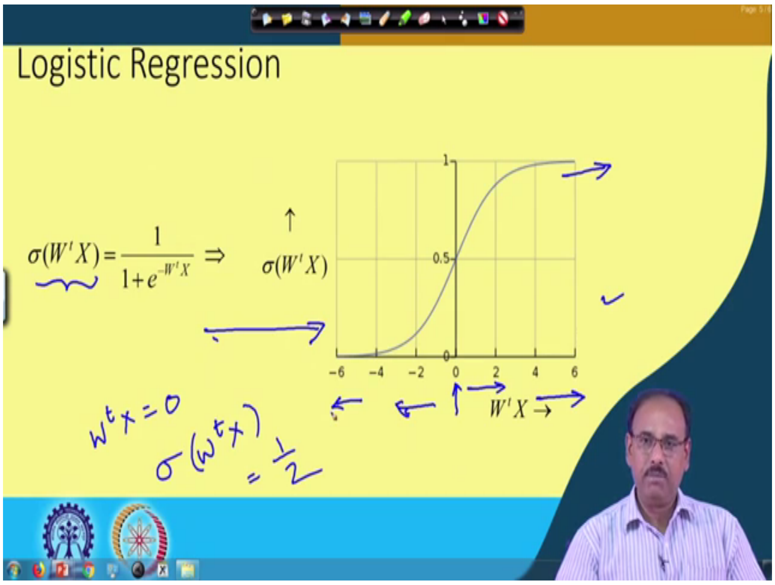
$$\underbrace{p(y \mid X; \overset{\downarrow}{W})} = \underline{\sigma(\underline{W^t X})}$$

So, in case of logistic regression I can have a probability measure that is I can compute that what is the probability of y which is the class index given a feature vector X and the model parameter W.

Which is written in this form p y given X and parameter vector W that can be written as a function of sigma W transpose X. So, again you find that this W transpose X what we said earlier is a measure of distance, this was a distance measure distance of X from the surface W transpose X equal to 0 and using this sigma function, I can convert this to a probability measure.

So, as we said that if my data vector X is far away from the separating plane my confidence level is very high. So that in other sense I can say that the probability of class y is very high. So, I can convert this distance measure into a probability measure and that is what is done by this function sigma.

(Refer Slide Time: 23:43)



Logistic Regression

$$\sigma(W^tX) = \frac{1}{1+e^{-W^tX}} \Rightarrow$$

$w^tx=0$

$\sigma(w^tx) = \frac{1}{2}$

And, I can write this function as a sigmoidal function which is given like this. So, this sigmoidal function is nothing, but sigma W transpose X equal to 1 by 1 plus e to the power minus W transpose X. And, if I plot this function the function will have a plot like this which is a sigmoidal function.

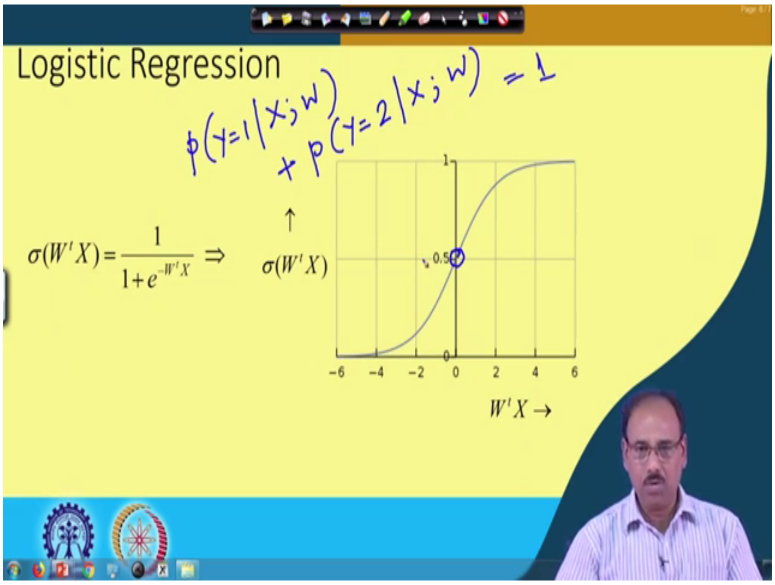
So, if I plot sigma W transpose X versus W transpose X I get a curve as shown on this side. And, here you find that as W transpose X become high as you increase W transpose X. So, coming to our previous interpretation of distance from the separating plane, as the distance from the separating plane is very high, the probability goes on increasing and asymptotically it goes to one. On the other hand if W transpose X is negative you find over here.

So, here you find that if W transpose X is equal to 0, what I get is sigma W transpose X that is equal to half right. So, from here e to the power of minus W transpose X if W transpose X is 0; that means, e to the power minus the W transpose X is 1. So, that I get simply the value of half. So, as at W transpose X equal to 0 the value of the sigmoidal function is half; that means, if the feature vector lies on the separating plane it has equal probability or belonging to class omega 1 and plus omega 2 or in other words I cannot really classify the feature vector X.

Whereas, as W transpose X becomes greater than 0 it is probability of belongingness to class 1 goes on increasing and when W transpose X becomes very high, that is the distance from the separating plane is very high, the probability goes towards 1. Coming to the other side as W transpose X becomes negative and becomes lower and lower that is it is more on the negative side, asymptotically your sigma W transpose X becomes tends to be 0; that means, it is probability of belongingness to class omega 1 is almost 0.

And, when I compute the probabilities because I have I am considering only 2 classes omega 1 and omega 2. So, I can say that the probability that it belongs to class omega 1 and the probability of it belonging to class 1 omega 2.

(Refer Slide Time: 26:30)



Logistic Regression

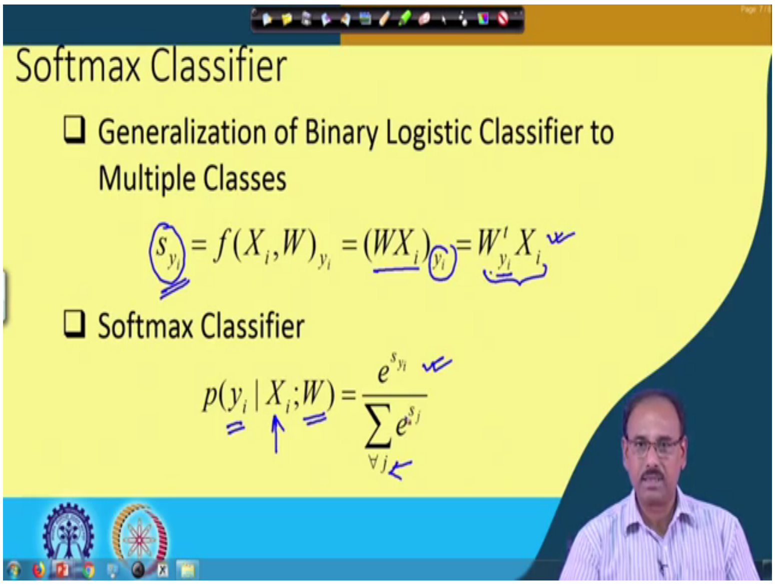
$p(y=1|x;w) + p(y=2|x;w) = 1$

$$\sigma(W'X) = \frac{1}{1+e^{-W'X}} \Rightarrow \overset{\uparrow}{\sigma(W'X)}$$

So, what I can write is I can write it in this form, that probability y equal to 1 given X and W, that is the probability that X belongs to class omega 1 and I can also write the probability that X belongs to class omega 2. So, probability y is equal to 2 given X and W. So, if I add these 2 that should be equal to 1 because I am considering only 2 class omega 1 and 2. So, here as the probability belonging to class omega 1 becomes high, the probability of belonging to class omega 2 becomes lower.

And, this is the point when both of them are equal that is probability of belongingness 2 both the classes are half to omega 1 as well as half omega 2. So, that is what is logistic regression, that is converting my distance measure or the confidence measure to a probabilistic measure that, what is the probability that X belongs to class omega 1 or what is the probability that X belongs to class omega 2?

(Refer Slide Time: 27:43)



Softmax Classifier

- ❑ Generalization of Binary Logistic Classifier to Multiple Classes

$$s_{y_i} = f(X_i, W)_{y_i} = (\underline{WX_i})_{y_i} = W^t_{y_i} X_i$$

- ❑ Softmax Classifier

$$p(y_i \mid X_i; W) = \frac{e^{s_{y_i}}}{\sum_{\forall j} e^{s_j}}$$

There is another concept which is known as soft max classifier.

So, earlier this linear regression or logistic regression, which we have discussed those are with respect to our binary classifier or 2 class problem. But, when we have a multi class problem, we have discussed earlier that I can have a linear machine, where using a linear machine, the parameters set of parameters are represented as a weight matrix W, while vectors in every row of the weight matrix W we have said represents a prototype or represents a class or a sample belonging to a particular class, that is the representative of a particular class.

So, if I have say k number of classes I will have k number of rows in the weight vector. And, what that set of our matrix of weight vectors give is when you multiply that with your given feature vector, you get a score vector. And, we have seen that the score vector is a set of real numbers.

So, if I have got k number of classes my score vector X was having k number of elements. So, if I have a feature vector X which belongs to class omega which belongs to class say why I the class index is why I, then the corresponding score is given by is of why I which have we have computed as W X i and the y ith component of that. Or this also you can write as W y i ith row the vector corresponding to y ith row take the transpose of that vector and multiply that with X I.

So, that gives you the score of class y i for feature vector X or X i which belongs to class y i, that is we already know because this is a training vector. So, again this is on the output of a linear function, as we have done in case of in case of logistic regression, I can also convert this to a probabilistic measure. So, I can say that what is the probability of class of y i given sample X i and your parameters W given by the parameter matrix or the weight matrix W. And, we can compute this as e to the power S y i upon e to the power s j.

Where the summation sum of e to the power S T in the denominator where the summation has to be taken over all j. So, you find that this also converts our score factors or the score values to different classes in a probabilistic measure that what is the probability of class yi given a feature vector X or even a feature vector X i. So, we will continue our lecture further.

Thank you very much.