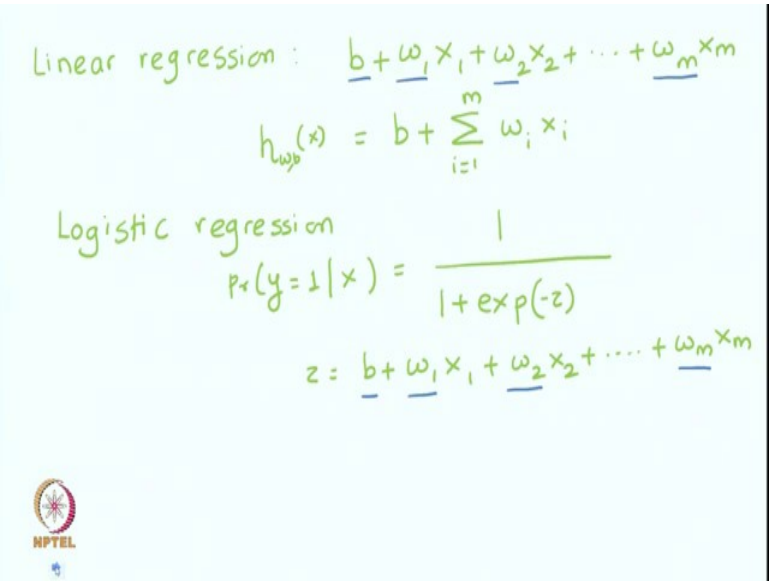
Practical Machine Learning with TensorFlow
Dr. Ashish Tendulkar Google
Department of Computer Science and Engineering
Indian Institute of Technology, Bombay

Lecture – 04
Loss Functions in Machine Learning

In the last session, we studied how machine learning model looks like, the session before that we also studied how the input data looks like. Now that we know about what is a training data in machine learning and also how model looks like, we will change gears to understand how to estimate the parameters of the model.

(Refer Slide Time: 00:53)



We looked at linear regression model and the linear regression model looked like: b + $w_1x_1 + w_2x_2 \ldots w_mx_m$. I can compactly represent this as: $h_w(x) = b + \sum_{i=1}^{m} w_i x_i$

This is my linear regression model and we normally use $h_w(x)$. So, this is the regression model.

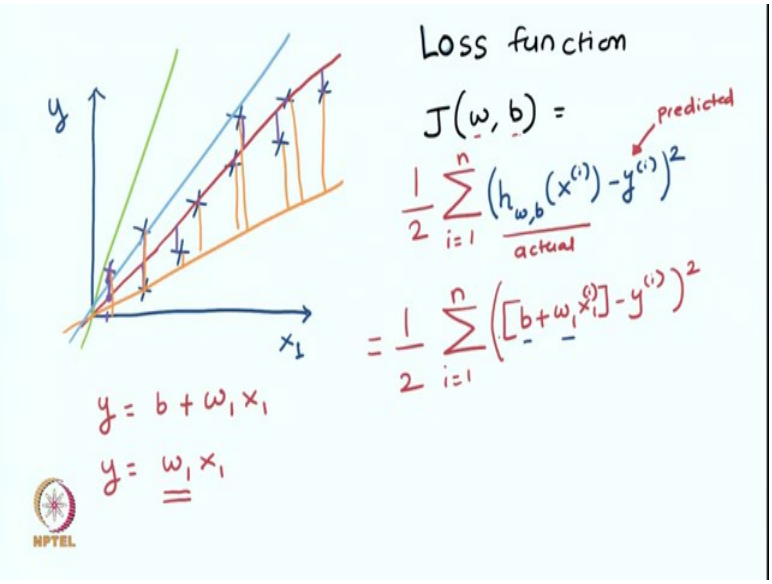
We also looked at the logistic, how does logistic regression model looks like? Logistic regression model predicts the probability of y=1:

$$p(y=1/x)=\frac{1}{1+\exp(-z)} \quad \text{where} \quad z=b+w_1x_1+w_2x_2+...+w_mx_m$$

And in case of neural network model we learned some complex function which also had lots of parameter with it.

Now that we have this particular model and we have trained data our job is to come up with values of these parameters. So, let us try to build an intuition about it based on a very simple example of a linear regression with a single variable.

(Refer Slide Time: 03:15)



Let us say this is a single variable x_1 and this is the value y to which is the output value or the label and let us say these are the points. Let us say we want to fit a line. Linear regression represent a line. So, let say we have a line which passes through the origin. So, the equation of this line is $y = b + w_1x_1$ here $b = 0$ because line passes through the origin. So, we have w_1x_1 as an equation of this line.

Now our job is to estimate the value of w_1, let us say if I use a different value of w_1 it will result into a different line. So, here we have a line which passes through the origin and which has got some slope. If I change the value of the slope I can draw let us say some other line from the origin. So just by changing the value of w_1 I can get different models or different functions.

Now, which of this function is the most appropriate function for our cause is the central question in front of us. So, we are having training data, we have fixed up our model and now the task is to estimate the model parameter and one of the tool that we use to estimate parameter model parameter is called loss function. We normally denote loss function with J(). Loss function is the function of parameter value, depending on what parameters we choose we get a model and because of model we incur some loss. Let us see what loss means in the context of linear regression.

You can see in the image for this particular value of x_1 this was a true value, but if we use red line as our model then this is the predicted value. So, this is an actual value and this is a predicted value for this value of x_1. So, we incurred some error. The error is difference between the actual value and the predicted value.

How do we measure this loss cumulatively? So, what we do is we find out the loss at every point and summit across all the points, let us write it mathematically how we do it for a single point.

$$J(w,b)=\frac{1}{2}\sum_{i=1}^{n}\left(h_{w,b}(x^{(i)})-y^{(i)}\right)^2$$

So, this is the loss this is the actual value if this equation is bit scary to you. So, you can read that this is an actual value, this is the predicted value and we square this up. So, we have actual minus predicted and we sum this loss across all end points and we add one half as a mathematical convenience. So, is this clear to you to everyone. So, what we are doing here is we are calculating loss at every individual point and then summing of the loss across all the points and this is the total loss that we incur and this is a total loss that we incur because we choose parameters w and b and because of w and b we get a model and because of model we incur some loss, so that is the relationship.

So, if we expand this in case of linear regression this will look something like this

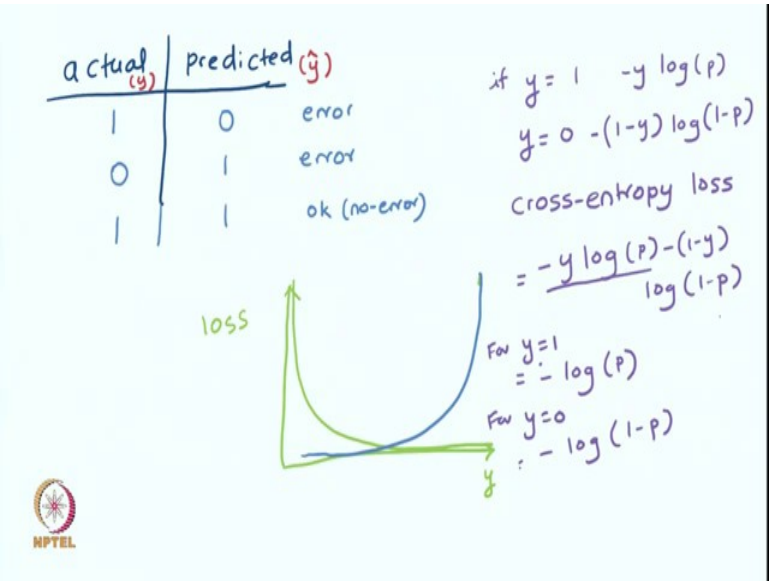
$$J(w,b)=\frac{1}{2}\sum_{i=1}^{n}\left(b+w_1 x_1^{(i)}-y^{(i)}\right)^2$$

So, you can now see it is pretty much obvious to see that this J or the loss function is actual function of the parameter values. So, loss function is the central piece that helps us to identify model parameters such that the loss is minimize. So, we try to identify

parameters in such a way that we minimize our loss and we will see how to minimize the loss in the next session. So, this is the loss that we compute for linear regression.

Let us try to see how we can formulate a loss function for classification problems. In case of classification problems we have two labels. So, let us let us prepare a table of the labels.

(Refer Slide Time: 10:10)



So, we have an actual label and we have a predicted label *y* and this ŷ. If actual value is 1 and if you predict 0 then there is an error or vice versa actual value 0 we predict 1 that is an error. If actual value is 1 we predict 1 that is fine.

So, let us try to develop an intuition for the loss in case of classification. If the actual value of y 0 and if you predict 1, you want to give a very large penalty and if actual value of y = 1 and if we predict 0, we want to give a very large penalty as well. So, we will we have very similar curve here.

We write this mathematically as if y = 1: -y log (p) and if y = 0: –(1 – y) log (1- p) and this gives us what is called as cross entropy loss which is written as:

$$loss = -y\log(p) - (1-y)\log(1-p).$$

So, let us try to understand when y = 1 what happens for y = 1 we get the term -log (p) and since y = 1 this becomes 0. So, eventually we only get the first term for y = 1. For y

= 0 you can see that the first term become 0 and we only get the second term, so we get - log (1 - p).

So, this is a clever representation of this two losses into a single equation. This cross entropy loss that we try to minimize while solving classification problems. So, cross entropy loss is specifically used for binary classification problems. For a multiclass classification problem, we use categorical cross entropy loss. If we represent our output or our labels as integers we use sparse categorical cross entropy loss. So, this is these are the loss functions that are used for classification task, whether with logistic regression or also with neural network models. So, having defined a loss function we know how to measure an error ones we fix up the parameters of the model.

Now, our job is to find out the optimal values of parameters such that the loss function is minimized. How do we really solve this problem and this is where optimization techniques or optimization algorithms help us in tackling this particular problem.