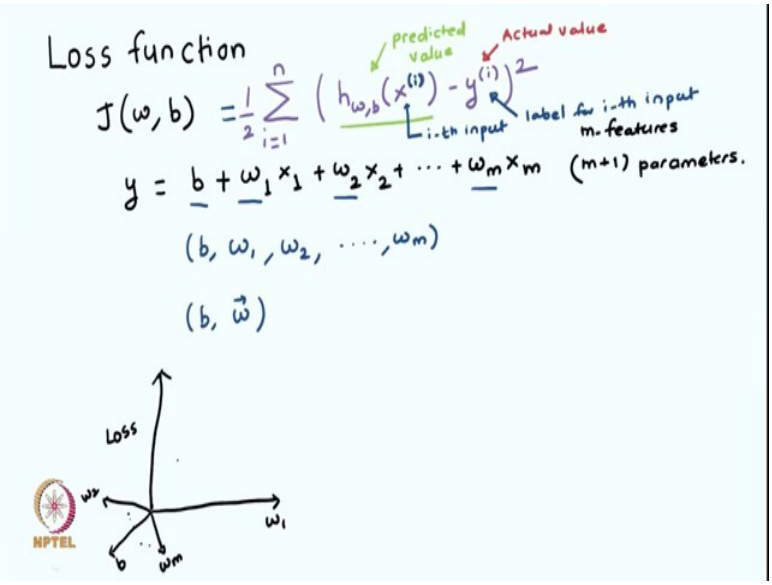
Practical Machine Learning with TensorFlow
Dr. Ashish Tendulkar Google
Department of Computer Science and Engineering
Indian Institute of Technology, Bombay

Lecture – 05
Gradient Descent

[FL]. In this session, we will study about optimization algorithms and their roles in machine learning. We will specifically study an optimization algorithm called Gradient Descent and its variation mainly: minimize gradient descent and stochastic gradient descent.

(Refer Slide Time: 00:45)



We studied what is called as loss function of machine learning model. We use symbol J w, b. So, loss function is parameterized by w and b, which are parameters of our model. So, just to remind you our model that we considered for linear regression was as follows: $y = b + w_1 x_1 + w_2 x_2 \ldots\ldots w_m x_m$

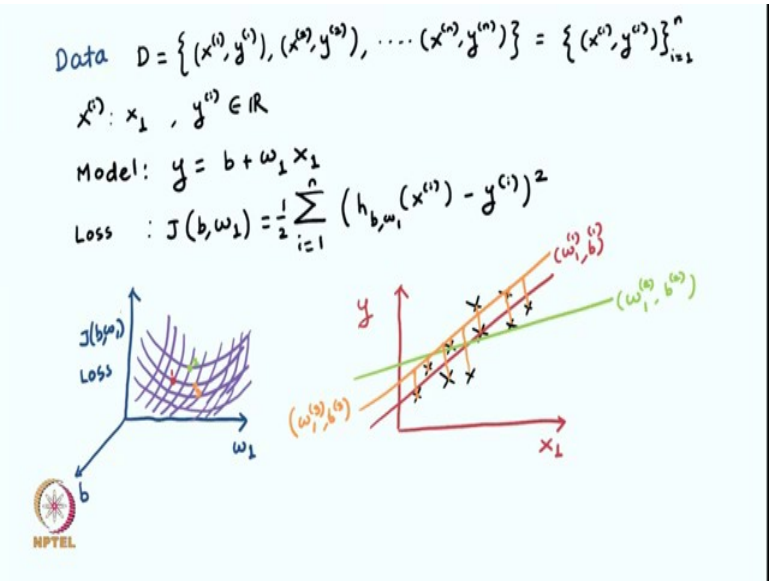
Here, we have training data with *m* features and we have modelled with *m + 1* parameters and these are the list of parameters. So, we can think of model as parameterized by b, w_1, w_2, ……w_m.

We will use w as a vector to represent, all these *m* weights. So, that is why we say that the loss is the function of parameters and in case of linear regression the loss function looks something like this.

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

Let us consider there are there are *m+1* parameters. Let us say this is the loss, this is parameter w_1, this is parameter b and there will be several such kind of parameters. Let us say w_2, w_3 ..... w_m. And loss will be a surface in *m + 2* dimensional space. So, it will be some kind of a hyperbola kind of a shape. In order to understand this; we will take a simplified model and try to visualize the loss function in the 2D space.

(Refer Slide Time: 05:24)



Let us take a simplified model where there is a single parameter. Our data is of the following form: So, we have data D is we have this pairs. We have the feature and the label. We have *n* such data points. So, we can compactly present this as:

$$D=\{(x^{(1)},y^{(1)}),(x^{(2)},y^{(2)}),....,(x^{(n)},y^{(n)})\}$$

We have ordered pair of features and labels, and there are n such kind of ordered pairs. Each data point has a single feature which is represented by x_1 and or y is a real number. So, this is a regression problem and the model that we use is $y = b + w_1 x_1$.

So, the loss function is

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

We compute the difference between the actual value and the predicted value is a predicted value minus the actual value and we take square of the difference. Geometrically, these loss function look something like this. So, we have J (b, $w_{1)}$ or the loss on y axis. We have weight w_1 on one axis and b which is a bias on the other axis.

It is a bowl shape function right. So, what will happen is we have to find out optimal point on this bowl. We have a model, where we have input x_1 and we want to predict value y. Each of this point on this surface if you take this particular point, this particular point; let us show some data points here these are data points.

If you use this particular point over here, this point would correspond to a line, this point is represented by two numbers which is w_1 and b and for w_1 and b it gives us a specific loss and our model also has two parameters which is w_1 and b. So, for this specific value of w_1 and b we get such a model. If we select another point on this surface, let us say this particular point, it will represent some other line. So, let us call this as *point 1* and let us call this as *point 2*.

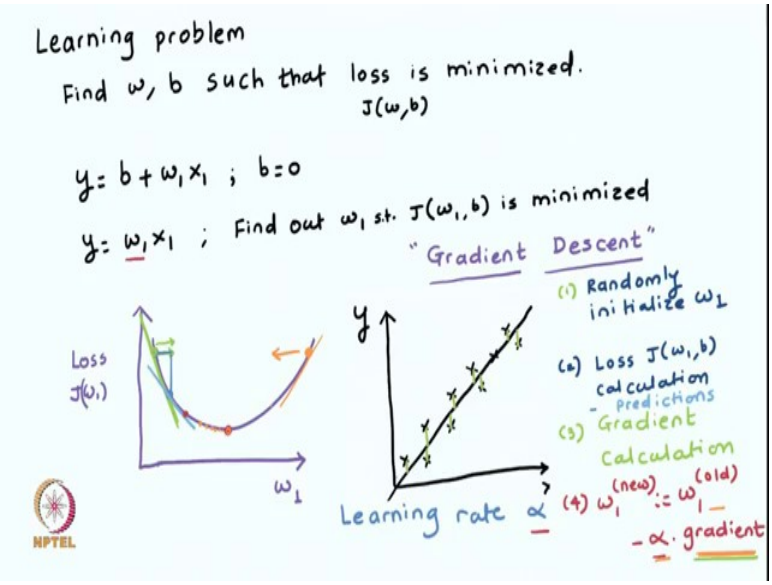
If you choose some other point it might represent some other line. Let us call this as point number 3. This is essentially a line which is ($w_1^{(3)}$, $b^{(3)}$). And for each one of them there is a loss and in order to recall what the loss is; so, for this particular line for the third line the loss is essentially a distance between the actual value and the predicted value. We calculate the difference and we take square of the difference. So, we find out all this differences and sum them up and that represents our loss.

If we sum all this numbers up that you will get a loss corresponding to ($w_1^{(3)}$, $b^{(3)}$). which is some number on the y axis which is a loss. This is very important to understand the duality in the loss space and in the model space. So, point in the loss function represents a model and we want to get a model that gives us the minimum possible loss. Our objective is to find a model or model parameters in such a way that the loss incurred due to those parameters is minimized.

You must be wondering that you can also try some kind of a brute force approach where you will explore this particular space and try to find out points that gives the minimum value of loss. But this is not really efficient. If we take let us say a parameter space or the loss function with m parameters with the value of *m* being very large and large *m*'s are kind of their routine in our day to day machine learning problems. So, if you are trying to solve this problem in the context of m which is some large number then you know a brute force is almost impossible. So, we cannot really do brute force. So, we have to do something more intelligent.

And you know the way we are phrasing this particular problem, we are saying that we want to find the values of parameters in such a way that the loss function is minimized. So, this is a minimization problem find w and b or find parameters such that the loss function is minimized. Let us see how to do that and let us first develop the intuition of it, and then get into the details of our first optimization algorithm which is gradient descent which is work hours of machine learning algorithms we will see that in a minute after understanding the intuition behind it.

(Refer Slide Time: 13:44)



So, what is our learning problem? The learning problem: Find w and b such that loss is minimized and loss we represent with J(w,b). Let us try to understand how we can do this intuitively. So, what I will do is I will again consider our linear regression model, y

is equal to b + w_1 x_1 and in order to give order to keep the expression simple we will assume that b = 0, so we get very simple model which is y = w_1 x_1.

Now, our job is to find out now the optimization problem is find out w_1 such that J (w_1, b) is minimized. So, there is exactly one parameter here because you have already said b = 0, so there is exactly one parameter. And now we will first visualize how the loss function looks like. So, loss function is parameterized the value of w_1. For each value of w_1 we get some loss, job done.

We have a mean squared error or a squared error as a loss function. So, you can see that it is a bowl shape function. This function in the language of mathematics is called as a convex function. So, what we will do is, so essentially what is happening here is for each of the value of w_1 there is a corresponding value of the loss, and you can visually see that this is the point where loss is minimized where there is a the value of loss is minimum.

So, since this is a problem with a single variable or with a single parameter we can visually find it. But the problem here is if we have multiple parameters, we cannot even visualize the loss function, how can we algorithmically or how can we programmatically find out this particular point. So, what is given to us is: we essentially know the loss function, and we want to find out the value of the parameter such that the loss is minimized.

Now, there is this particular method which is called as gradient descent that helps us to do this programmatically. Let us try to understand gradient descent intuitively before getting into the details of the steps involved in the process. We first initialize the value of w_1 to some random value. So, for this particular value of w_1 there is a loss associated with it. So this is the point where my initial guess landed. So, my initial guess is this for the value of w_1 and at this point what we do is we calculate the loss.

So, what is it representing? I am selecting, I am randomly setting the value of w_1 to some parameter. This will actually define a model for me. So, remember the duality of the loss space and the model space, so we have x_1 here and we have y here. Let us say these are the points and we have model which is a line passing through the origin.

This is a difference between the predicted values and the actual values. We calculate square of the difference and sum them up across all the points and get loss corresponding

to this particular model. Now, so the first thing that we did is we randomly initialize w_1, then we calculate the loss value. Now, this is the point where we want to reach.

How do we really reach this point from here? Now, think of this and as a task that is analogous to, let us say climbing down from the mountain top. So, what happens is while we are climbing down the mountain top, we are let say at a specific point, we look around and find out what is a direction that will take me down to the valley. So, in gradient descent we exactly do it at a particular point on the loss surface.

So, at this point what we will do is we will calculate the slope or the direction in which I should be moving. So, let us say this is a slope, this point. So, this is a tangent to this point. So, we can calculate slope of this tangent. And we get the direction of the slope. So, we will move in the direction that is opposite to the slope. So, the slope is negative here, we will move in the negative directions. So, first is we calculated the loss. Second, we calculate the gradient. And once we know the gradient, the next question is how much we move from the original point so that we reach the valley.

So, there are multiple options that we have. We can move or we can step, we can have a longer strides or we can take shorter strides. So, the length of the stride is decided by a parameter called learning rate, which we denote by α. So, learning rate helps us to control how long strides are we going to take from a particular point. Let us say if we are at this point and if we have some learning rate, we are going to take a stride and we will end up over here. So, this becomes our new point.

And what we do is we repeat the same process that we did at this point at this point, we first get the predictions on this particular model, we calculate the loss and then we calculate the gradient. In order to get the loss, we should also have predictions. We need to get predictions at every point. As you can see here this is a prediction for this particular point, so predicted value and actual value; because in order to calculate loss we need to know what is the actual value and what is the predicted value.

So, we first make the predictions with the value of the parameter. We substitute that in the model and we do the predictions, and based on predictions we calculate the loss. And after calculating the loss we calculate gradient of the loss. So, we will again do the same thing at this particular point and we see that it is a direction of the slope. So, we will be moving in this direction by taking some step. So, let us say we come here. Now, you can

see that as we approach this particular point which is the point whether loss has got the minimum value, as we go closer and closer to that point the derivative or the gradient will become smaller and smaller.

At this particular point the gradient will be 0 because it is a minimum point. So, you can see that as we as we move closer to the minima, the gradient value will become smaller and smaller. So, we have a constant learning rate. So we calculate gradient, we have learning rate. And then what we do is we have new point. So, let us say

$w_1^{(new)} = w_1^{(old)} - \alpha$. gradient

So, we can see that when learning rate is constant, gradient becomes smaller and smaller.

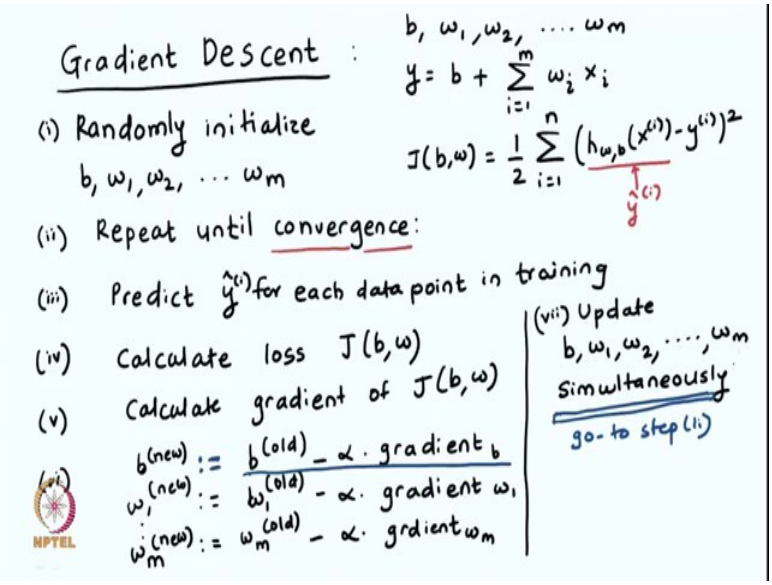
So, we will be making eventually our stride will become shorter and shorter as we approach the actual minima. So, this is how this is how we reach to this particular point. Now, you can you can work yourself, you know you can take a pause here and you can work out yourself how this particular calculation work if I randomly initialize the point over here. You can see that the gradient, this is the direction this is the slope and we will be moving in the opposite direction of gradient.

So, we will be since gradient is positive here, we will be moving in the opposite direction. So, what will happen is since gradient is positive at this point, we will be effectively moving in the opposite direction because this is the value of $w_1^{(old)}$, we are going to subtract something from it. So, we will be moving in this particular direction, if I am starting from here.

On the other hand, when we started from here, here the gradient was negative and since we are going to move in the negative direction of this. So, you can see that gradient is negative, this negative and negative becomes positive. So, we move in the positive direction or we have some value we are adding something to it, so we are getting a value which is greater than the old value. So, if we start from here we are going to get the value of w_1, which will be greater than the previous value. If we start over here we will get values of w_1 which will be lesser than the previous value. So, this is the intuition behind gradient descent.

Let us write down steps in gradient descent.

(Refer Slide Time: 28:59)



We will try to generalize the gradient descent for m parameter case. So, when we are trying to establish the intuition of the gradient descent we did it intentionally with a single parameters, so that it is easy to geometrically show what is happening. But when we go to the to the m parameter setting it becomes difficult to visualize what is happening.

Let us consider a setting where we have m plus 1 parameters in the model b, w_1, w_2 ... w_m, and we are trying to solve for a regression, so the model is:

$$y = b + w_1 x_1 + w_2 x_2 \ldots\ldots w_m x_m$$

So, this is a linear combination or linear combination the parameter and the feature value. And this is short form, these are short hand form of writing the model, and obviously, our loss function is:

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

So, the first step is we randomly initialize b, w_1, w_2 ... w_m.

So, we randomly initialize all the parameter values. Then, we are going to repeat until convergence. We first use all this parameter values and we predict ŷ for each data point

in training. So, we calculate $\hat{y}^{(i)}$ or the predictions first, then we calculate loss, we calculate loss which is J (b, w).

We know that the predicted value and we know the actual value based on that we can calculate the loss. Then we calculate the gradient of the loss. We will we will see how to calculate gradient of loss. Fifth step is

$b^{(new)} := b^{(old)} - \alpha \, . \, gradient_b$

So, $gradient_b$, $gradient_{w1}$ and we do it for all the parameter values. Update b, w_1, w_2, w_m simultaneously.

We are calculating this gradients with respect to each of the parameter values. notice that this is not an equal to sign, we are using some kind of us other notation where the effect of this is we are setting the value of b to a new value which is coming from the equation on the right hand side in this case which is this equation. And when we are calculating gradient with respect to w_1, we are not going to use this $b^{(new)}$, we will still be using the value $b^{(old)}$ and all the old values for all other parameters. And we change the values from old to new right at the end in the step number 7. So, this is what is called as simultaneous update.

So, this is important to note that all this parameters have to be updated simultaneously at the end of the loop. And then you again go back to this two, we check for the convergence, and we essentially repeat. After this step is we get a new point on the on the loss surface and we repeat the same process at that particular point in the loss surface.

So, this is a sketch of gradient descent algorithm for your reference. We have not yet talked about how to determine the convergence of this or how to calculate the gradients which we will see in the next session [FL].