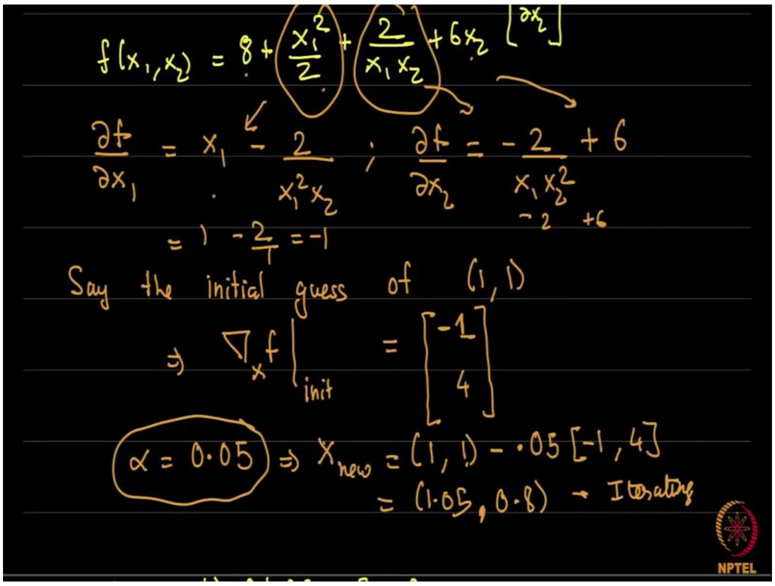
Inverse Methods in Heat Transfer
Prof. Balaji Srinivasan
Department of Mechanical Engineering
Indian Institute of Technology – Madras

Lecture – 22
Gradient Descent - Simple Example

(Refer Slide Time: 00:19)



Welcome back. In the last video we looked at the gradient descent algorithm and in we also took a simple example last time, but let us look at another simple example and I will show you a code also for this very simple example in this case. And in the future videos we will see how to apply gradient descent actually to inverse properties. So, let us look at this simple case here.

Suppose we want to minimize this function. So, in this one in this case it is kind of hard for you to know the analytical solution though and you can actually theoretically calculate it. But the way to find out of course is to do a derivative of f with respect to x_1 set it to 0, $\frac{\partial f}{\partial x_2}$ equal to 0 this will give you the theoretical solution. But when we want to do gradient descent to find out the minimum we start with some arbitrary guess, in this case the arbitrary guess has been given to be (1, 1) with initial guess 1, 1.

So, let us call this X_0. So, just to be consistent let me call this capital X_0. So, capital X_0 is has to sub. Now we have the general formula as you remember,

$$X_{i+1} = X_i - \alpha \nabla_x F$$

In this case with respect to x and this turns out to be,

$$F = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix}$$

Now this is simple enough but we have to calculate this. So, given the function f of x or $f(X_1, X_2)$ is this function here is,

$$f(X_1, X_2) = 8 + \frac{{X_1}^2}{2} + \frac{2}{X_1 X_2} + 6X_2$$

You have to find out gradient of f is composed of $\frac{\partial f}{\partial x_1}$ which of course is this function will simply give,

$$\frac{\partial f}{\partial x_1} = X_1 - \frac{2}{{X_1}^2 X_2}$$

and this function of course will give you nothing. Similarly, $\frac{\partial f}{\partial x_2}$ this gives nothing this gives nothing the third one gives us,

$$\frac{\partial f}{\partial x_2} = \frac{-2}{X_1 {X_2}^2} + 6$$

So, just as a sample calculation um say the initial guess not same but we are given an initial guess of 1, 1 in that case,

$$\nabla_X f|_{init} = \begin{bmatrix} -1 \\ 4 \end{bmatrix}$$

So, let us say Alpha equal to 0.05 this will give us,

$$X_{new} = (1,1) - 0.05[-1,4]$$

So, this is,

$$= (1.05,0.8)$$

So, you can keep on iterating. So, I just took some arbitrary value of alpha it is possible to play with more values of alpha. So, I will just write this down as a code and you can see how this performs for various values of alpha. So, let us go and see the code now.

(Video Starts: 04:49)

So, here we see our code as usual I have kept it in a MATLAB live file. so, that we can intersperse text with code. So, if you can see the question repeated here just to remind us of what we are looking at we have we want to minimize this function $8 + \frac{{X_1}^2}{2} + \frac{2}{X_1 X_2} + 6X_2$ and the way we have done it is first we specify the function and then we specify the gradient. So, here you have it, I have specified the function $8 + \frac{{X_1}^2}{2}$, this I hope everybody recognizes is a $+ \frac{2}{X_1 X_2} + 6X_2$.

So, this is the function here the gradient was exactly the thing that we already calculated just a little bit before in the video $X_1 - \frac{2}{{X_1}^2 X_2}$ and the second component the semicolon here tells you that is the first component and this is the second component$\frac{-2}{X_1 {X_2}^2} + 6$. So, we have written the gradient here number of epochs simply means the number of iterations that I wish to do.

So, right now I have given this as 5 and I pre-specified the learning rate as 0.05 and you here you can see these are some variables which are useful for plotting you do not really need it all you need is just this. In fact, I can remember remove this line entirely and everything will still work this is just for some plotting that I am going to do later on below. So, let us now start running this program.

So, as you can see after the initial guess the value of w is (1, 1) and after one iteration this is exactly what we had calculated theoretically you can see this 1.05 and 0.8. So, hopefully you can see that but of course it has done more iteration. So, you can continue your iterations 1.1109, 0.6488 so on and so forth and I have visualized. So, you see this plot here we started at this location one, one and we started coming below.

So, it has started reducing significantly and as I told you earlier typically the idea is that the orange and the yellow portions are higher values of the function, you can see the level which shows here on your screen is 22.13 to come a little bit below it is 21.49 and you can keep on seeing that this is reducing. So, same here at this location 1, 1 you would be at a slightly lower point and you are going to come down.

But of course, we have not reached the real minimum here, which is somewhere around this point that I am demonstrating, as you can probably see, somewhere to the center of this seemingly elliptical region is our minimum level. So, somewhere there is our minimum level. So, all that means is we need to increase the number of epochs but before we go there, I want to show you what happens if we increase Alpha.

So, suppose I increase Alpha you can see that the behavior starts becoming a little bit more erratic if I increase Alpha further you see the Contour is left here it has gone to some really bad values. So, it converts really bad if you increase alpha or if I even decrease Alpha to point one and let us say increase the number of iterations to 10, we can see what happens. So, it is actually not staying at the minimum and it starts moving out and this is what is known as you know divergence.

And we need to do some hyper parameter optimization. Now I had done this beforehand I knew that 0.05 would work which is why I demonstrated it here but there is no easy way of finding this out beforehand. you have to play with the parameters. So, now I have made the number of epochs 50 and you can see it is sort of seemingly converges but maybe it diverges just a little bit we can make alpha a little bit smaller still let us say 0.02 and increase the number of epochs to let us say 500 and see if that works fine.

And you can see that it is kind of converging right to the minimum and somewhere below here at the bottom you will see the converged value it is you can see. Now the gaps between any two subsequent values at least to four decimal places we have not changed. So, you can see that for a large number of iterations we are stuck here at least for four decimal places of iteration. So, we can be satisfied with the minimum for this function is somewhere around 1.6438 and 0.4503 and you can check theoretical minimum is indeed somewhere around that.

So, this is a very simple example of how you can program gradient descent just as long as you have the formula for the gradient you do not need anything as complicated as the normal equations you can see this is a simplification. So, this is very straightforward. Now whether the function is linear or non-linear and in fact you can see here the function is non-linear in the parameters remember we are minimizing with respect to x and even for non-linear parameters this function works just fine.

So, we can experiment with this function with the non-linear case that I introduced to you at the beginning of this week, which was the non-linear regression with our temporal unsteady problem and we will try and do that next and see what happens, thank you.

(Video Ends: 10:38)