Chapter 5.2 - Derivative Based Optimization

15 MIN · YOUTUBE · HTTPS://WWW.YOUTUBE.COM/WATCH?V=7U_HTAAMXES
https://www.youtube.com/watch?v=7U_hTAAmXes

SUMMARY

The discussion focuses on optimization techniques using derivatives, specifically gradient descent, to find minima of functions. It explains the importance of derivatives in determining the direction of change in a function and how gradient descent iteratively moves towards a minimum by following the negative gradient.

- *Gradient descent is a fundamental algorithm for finding minima of functions using derivative information.*
- *The derivative indicates the slope of a function, and when it equals zero, it signifies a potential minimum or maximum.*
- *The second derivative helps determine whether a critical point is a minimum (greater than zero) or maximum (less than zero).*
- *In higher dimensions, the gradient generalizes the derivative, indicating the direction of steepest ascent.*
- *The algorithm iteratively updates the position by moving in the opposite direction of the gradient, effectively "walking downhill."*
- *The step size, or learning rate, can be fixed or optimally calculated, impacting the convergence speed.*
- *Gradient descent can converge to local minima, and techniques like stochastic gradient descent can help escape these to find global minima.*
- *Practical applications include curve fitting, where initial guesses significantly influence the optimization outcome.*

we're going to return now to optimization but now we're going to make use of the derivative remember from calculus in some sense what a derivative is is a tangent on a function and it sort of gives you a slope or a direction and often what we want to do with this is use derivative information to find which way is are we is sort of in some sense a towards the Minima so we're going to aim for thinking about these

methods and using the derivative information to help us find Minima we have no constraints on this optimization so this unconstrained optimization but using derivative information and really the heart of this is what's called gradient descent so the gradient descent algorithm is one of the most important algorithms that we use because it really starts to it's a it's one of the most standard ways to sort of find functional forms or Minima functions so remember we start off with an

objective function and that objective function is just basically find the minimum of this F ofx so this is what we want to do just a reminder about derivatives let's first of all this is X generally with the control variables can be high dimensional can X is a vector of you know n components but if we constrain ourselves for now to a one-dimensional function f ofx then what we want to do is find the Minima what we know the Minima has is when the

derivative is zero this is one of the first things you learn in calculus right is what does it mean for the derivative to be zero that means you're either at a Minima or a Maxima so this condition becomes very important for us to start thinking about more precisely in calculus we learned that if you take the second derivative and if you take the second derivative instead it's zero or sorry and if it's greater than zero when the derivative is zero then that

means you're at a minimum and if it's less than zero you're at a maximum so we are actually at this way in this form setting up looking for a Minima by the way looking for a Maxima is just the opposite you just take the function and multiply by negative and the Minima is now the maximum or vice versa maximum is now the Minima so all of it can come down to looking for a minimum of a

function so remember now though we are really looking in a higher dimensional space and so we need are methods that allow us to search over all this space here of these variables in order for us to find where this Minima is most of the methods we're going to talk about are going to get us towards finding a local Minima a global Minima is a much more difficult task and generically we don't have good methods for doing that overall

however we do have methods that can allow us to get out of local Minima towards going towards the global Minima if we in many ways especially in high dimensional spaces this is what's going to be called stochastic grading descent and we will cover that later on but for now what we're going to do is now that multi-dimensional we're going to make use of the gradient here's the gradient so it's normally it's a gradient F ofx and what we're looking

for for the Minima is when this is zero so this is just a generalization of the derivative it equal to zero if we find in each Direction the gradient is basically taking the derivative at each of these different control variable directions when that's zero you have found the minimum all right so here's a simple example of a function here's the function x^2 + 3 y^2 so it's this parabolic shaped bowl and the Minima here is at zero so you can see right

there there is the Minima point and everything and so one of the nice things this is actually a convex function so it means what we're going to do is when we think about there's only one Minima and it is the global Minima so often convex optimizations are much much easier to do than when you have much more sophisticated functions where you have multiple Minima but here we have one Minima so what we can look at is if we

want to compute the gradient along every point or even at a given point suppose we pick a point right here and if we compute the gradient it basically is giving us the direction of the most rapid change so that's going to be this Arrow here so it's telling you which way is the maximum Direction locally at that point of going uphill and so if we draw this line here right this is what the gradient is finding for us in some sense

this is our tangent line that we want on that function and so what we're going to do is that is the direction where you're having the maximum amount of change and the gradient gives you the uphill Direction so what we're going to do is walk downhill in the gradient descent so by the way here are these arrows that show you gradients computed locally at a bunch of different points and you can see which way they're pointing which is

in some sense the direction of Maximum change away from the origin here and so what we're looking at is here is these directions and what we're going to do is step in the opposite direction towards downhill okay how do you compute the gradient well here is the formula for the gradient at least for the 2D function it's the derivative respect to X in the X Direction derivative respect to y in the y direction so this is

telling you how it's changing not in just one dimension but in two and for this function this is pretty easy to compute it's computed right here right so it's 2x in the X Direction 6 y in the y direction so that gives us our gradient Direction at every single point and so what we're going to do is set up a scheme or an iteration procedure all these optimizations are based on iteration procedures where you walk downhill and here how it's set up so if

you look here what we're going to do is we're going to get a new value this is going to be the new value we're going to start from the current location we have which is is X and notice what this is It's the gradient of f in other words the gradient telling you way the maximum Direction things are changing uphill and we're going to go in the opposite direction that's why I have the minus sign and to tells you how big a step do

you take going downhill okay so that's going to be our updated value of where we're going to step so we're going to walk downhill by a an amount Tow and the idea here in an iteration scheme is once I've taken that step downhill then I evaluate at the next point I'm at which way is the gradient telling me I should step in other words which way is the maximum direction of change going downhill so the one question still left

over then because you can compute the gradient is how do I determine this towel so what we're going to do here is you'd like to take in some sense an optimal towel an optimal step downhill and the minute I say you take an optimal step you know it's going to involve a derivative calculation so what we're going to do is think about this F here as a function f of C in other words F of this new value that we're taking which

is C of T which is the next step and we're going to take a derivative of this with respect to to so if you use your chain rule here's what your chain rule gives you it's gradient of f which is function of C gradient of C which gives you just if you take the gradient C you just get back here F itself and you're setting that to zero in other words that's going to going to give you your

optimal size tow if I solve this equation so let's do it here for this example here I have f ofx is $x^2 + 3 y^2$ you can easily compute this update rule C is equal x minus t gradient F so if you write this all out here it is and now what we have to do is compute this F of to which is given right by here so if you compute the F of to you get this

formula here okay and this is just based from here on that equation and now what we do is take the derivative of this with respect to towel set it equal to zero and this is what you find so this towel here is the optimum step size going downhill okay so we're going to do those steps as we iterate along so first we compute the gradient second we compute the step sizes and now we're ready to build an algorithm that marches us

downhill and so here is a simple example of this in Python so if you notice of the way we set up the code structure so first of all we set up an array we're going to keep track of our steps as we go we're going to start off with some initial guess that's what the xnot 3 and Y KN 2 is that's my I picked the position to start off as 32 that was my initial guess and the value of the

function there is given right by here and now we're going to do is go into a loop and the first thing we do is compute the step size that we're going to take compute the update rule because we already know what the gradient was so we can just evaluate the gradient in the X Direction and the y direction update them and then compute a new value of f now if the value of f here is within

some tolerance in other words if I'm close to zero I should stop and then I break out of this Loop so we're going to just get iterated along this Loop until this condition is satisfied and so this is gradient descent walk downhill with optional step Siz it's a very simple function here is in fact the solution of this so what you're W looking here is in the XY plane I started at this location here I took an

optimal step which got me to here took another optimal step here and so forth and so if you look here at the error in terms of what my distance from finding the zero is it's dropping down monotonically all the way till here and after eight iterations look I'm already around 10- 6 that's with that Optimum step size now I had to pay a price for that because I had to compute at each update a new towel and that can be

expensive in practice when it's very high dimensional what's often done in practice is you don't actually compute the towel you just simply pick the step size you want to take so this towel is often called The Learning rate in machine learning and so we're going to just fix a towel so for instance here if you fix tow to be 0.1 you can see here this is the iteration procedure I am walking downhill I will find that Minima

but it's just a little slower because I'm not taking optimal step sizes and so here's the error as it drops down so again you can still get to the end result and of course I can adjust Tow and maybe make this a little faster maybe make it slower and so adjusting this learning rate as we go along is an important part of the gradient descent algorithm but in either case what the gradient descent is doing and it's one

of the core algorithms we need to understand is from a local position that I'm currently at which way is the direction that I want to take that is the maximum Direction which lowers some objective function okay so that's the downhill we talk about we're walking downhill in the objective function and so then we pick a towel to take a step in that direction either optimally or just fix it like this and we walk forward and again this is an important

piece of information you need and we're going to use gradient descent quite a bit as we go in in in this as we get into machine learning so there's also commands built in to python where we can look for Minima and so this is sort of a little bit of Beyond gradient descent so what I showed you there in gradient descent was sort of the simplest way to do gradient descent but as you can imagine much re much research has gone

into making improved optimization algorithms and and so we're going to and many of these are built into python or mat lab or whatever code you want to do and so let's talk about one here let's do a curve fit so the Dots here are you know data points and what I want to do is fit this curve to it FAL a cosine BX plus C so I want to just go ahead and try to fit that so in other words I have

to optimize over ABC to find the best fit so that is my objective okay is to fit this curve in a least Square sense where I have a B and C are the parameters I'm actually trying to find their best values so here's a simple code where what we're going to do is use What's called the minimize command in Python so I Define what I'm trying to fit and what I'm trying to fit is right here I'm trying to Fe a Leist least

square error which is the E2 error which is my curve which is a cosine BX plus CX which is basically here to the actual data square root divided by the number of points points so that's what I'm trying to fit here's the data itself and then what I do is I typically guess initial values for a b and c so if you look at that curve these are reasonable initial values to pick I don't know what

they are exactly but that's what I'm going to determine so I guess some good initial values and then I use the minimize command to go look at the function I'm trying to fit with the initial with the initial guesses and then I hear the arguments I'm using and what it's going to use is called a nelder meat algorithm which is some kind of iteration algorithm scheme which is sort of beyond what what's the standard gradient descent but very

much in line with what gradient descent does okay which is finds an iterative procedures to walk downhill and when you do that you get the fit I showed you which is this curve here now interestingly enough if you don't guess well if you if I just give you random guesses that are not very good you often will find a minimal solution with which is another local Minima somewhere else and it doesn't give you a good fit at

all so it's really important in gradient descent or even this minimize function that you provide a good guess otherwise this algorithm will just simply converge to a local Minima and it may be a local Minima very far away from what you would to have that makes sense for the problem so this is sort of your stereotypical method to use if you going to use derivative information which is gradient

descent it's one of the key optimization algorithms you need to know and you can see how it works just find the derivative take a step in along the derivative Direction and in fact this is exactly what we did with Newton rafson interation so it's a very powerful technique and heavily used across sciences and engineering and we will use it later especially in context of machine learning