Minimizing a Function Step by Step

We will start with optimization, it's a important part of deep learning.

So first, let's get the basic facts about three terms of a Taylor series. It's seldom that we would got up to third derivatives in optimization. So that's the most useful approximation to a function.

$$F(x+\Delta x) \approx F(x) + \Delta x \frac{dF}{dx} + \frac{1}{2}(\Delta x)^2 \frac{d^2F}{dx^2}$$

Here we are thinking of $F(x)$ as just one function, and x as just one variable, but now we want to go to more variables. So what do we have to change if F is function of more variables? $x = x_1, \cdots, x_n$

So just to get the words straight so we can begin on optimization.

$$F(x+\Delta x) \approx F(x) + (\Delta x)^T \nabla F(x) + \frac{1}{2}(\Delta x)^T H(\Delta x)(\Delta x)$$

$H_{jk} = [\frac{\partial^2 F}{dx_j dx_k}] = H_{kj}$ -- Hessian matrix.

Let's see vector function.

$$f = (f_1(x), \cdots, f_n(x))$$

$x = (x_1, \cdots, x_n)$. f is exactly we have in gradient. Think of these two as parallel.

$$f \approx \nabla F$$

so we can get

$$f(x+\Delta x) = f(x) + J\Delta x$$

J is the Jacobian matrix $J_{jk} = \frac{\partial f_j}{\partial x_k}$.

OK, so that's the background. Now, we want to look at optimization. We want to minimize the $F(x)$ and solve $f = 0$ (means $f_1 = 0, \cdots, f_n = 0$) (n equations and n unknowns). Let's first start the second one. And we will start with Newton's method.

Newton's method is a big application of gradients in Jacobians.

$$0 \approx f(x_k) + J(x_k)(x_{k+1} - x_k)$$

Let's use k for k times iteration. So we are at a point x_k. We want to get to point x_{k+1}. We write the above formula.

$$x_{k+1} \approx x_k - J(x_k)^{-1} f(x_k)$$

So that's Newton's methods for a system of equations. It's natural. And we are going to write down Newton's method for minimizing a function. This is such basic stuff that we have to begin here. Let's even begin with an extremely straightforward example of Newton's method.

Suppose we only got one function.

$$f(x) = x^2 - 9$$

and we want to solve for $f(x) = 0$. So what is Newton's method for it?

$$x_{k+1} \approx x_k - \frac{1}{2x_k}({x_k}^2 - 9)$$

We followed the formula. And let's simplify it.

$$x_{k+1} \approx \frac{1}{2}x_k + \frac{9}{2x_k}$$

We can check $x_k = 3$ and x_{k+1} stays at 3. The really important point about Newton's method is to discover how fast it converges.

$$(x_{k+1} - 3) \approx \frac{1}{2}x_k + \frac{9}{2x_k} - 3$$

Let's looking at the speed x approach to 3. Is it converging? How quickly is it approaching ? These are the fundamental questions of optimization.

$$(x_{k+1} - 3) \approx \frac{1}{2}x_k + \frac{9}{2x_k} - 3 = \frac{1}{x_k}[\frac{9}{2} + \frac{1}{2}{x_k}^2 - 3x_k]$$

$$(x_{k+1} - 3) \approx \frac{1}{2x_k}(x_k - 3)^2$$

That was the goal. That's the goal that shows why Newton's method is fantastic. the equation says that the error is squared at every step. So if we are converging to a limit, it will satisfy the formula.

Let's move on to $F(x)$. Minimizing $F(x)$ is like solving $\nabla F = 0$. And this is sort of the heart of our applications to deep learning -- we have very complicated loss functions to minimize, functions of thousands or hundreds of thousands of variables. So that means that we would like to use Newton's method, but often we can't. So we need to put down two methods -- one that doesn't involve those high second derivatives and one Newton's that does.

So method I. And this will be **steepest descent**. That says that

$$x_{k+1} = x_k - s_k \nabla F$$

steepest descent means that moving in the steepest direction which is the direction of the gradient of F. And we move some distance -- s, and we better have freedom to decide what that distance should be. s is a **step size** or in the language of deep learning, it's often called the **learning rate**. It's natural to choose s_k so that you minimize F. Imagine $x_k - s_k \nabla F$ is a line or a direction. the value of F will drop initially. But then at a certain point, it will turn back on you and increase. So that would be the natural stopping point. We would call that an **exact line search**. So exact line search is the best s. Of course, that would take time to compute, and you probably, in deep learning, that's time you can't afford, so you fix the learning rate s. so that's method one, steepest descent.

Now, method II will be Newton's method.

$$\boxed{x_{k+1} = x_k - H^{-1}(\nabla F)}$$

We are getting really the right direction using Hessian matrix. So the Jacobian of the gradient is the Hessian and that makes sense, because the first derivative of the first derivative is the second derivative.

Now, we have to think about solving these problems, studying the converge. The convergence rate for Newton's method is quadratic . The error gets squared, that means super-fast convergence if you start and close enough. The rate of convergence for a steepest descent is of course so good as Newton's method. So a linear rate of convergence would be right. You would like to know the error is multiplied at every step by some constant below 1. That would be a linear rate. Levenberg-Marquardt is sort of cheap man's Newton's method. It does not compute to Hessian.

We will move on to **convexity**. Convexity is the key word for these problems. Let's start with convex set -- K.

Here is our general problem, our convex minimization(you hope to have, and in many applications, you do have): minimize a convex function F for points in a convex set K. So that's like the ideal situation to get something on your side, something powerful , convexity.

the union of convex sets are usually not convex such as the union of two triangle is not a convex set.

the intersection of convex sets is always a convex set.

The coolest way is to connect the definition of a convex function the definition of a convex set. So a function is convex when the points on and above the graph are convex set. It's not our usual test for convex function. So now we want to give such a test.

The definition of a smooth convex function / How to test a function is convex. For one variable function, the second derivative is ≥ 0 will mean convex. Suppose $f(x)$ is a vector function. $f(x) = \frac{1}{2}x^TSx$. the test is the Hessian matrix is positive definite or semidefinite

The maximum of some convex function is again convex, but the minimum is not convex.