Steepest Descent Algorithm

Natasha Lee

Abstract

Partial derivatives, the Hessian matrix, and gradient descent provide a foundation for the steepest descent algorithm used to optimize certain outcomes in machine learning.

1 Partial Derivatives

1.1 Definition of Partial Derivatives

Partial derivatives are defined as derivatives of a multivariable function when all but the variable of interest are held constant during the differentiation [6]. In ordinary derivatives of single variable functions, $f'(x)$ or df/dx is the rate of change of the function as x changes. With multiple variables, we may want to only see how the output changes when only one variable changes or we may want to see how the output changes when multiple variables change [3]. In the case of the latter, there are infinite number of ways the variables can change. To denote the change of one variable while holding the others fixed, the notation is

$$\frac{\partial f}{\partial x}$$

where x is the variable allowed to vary and read as the partial derivative of f with respect to x. The partial derivative with respect to one variable is then differentiated in the same manner as an ordinary derivative. In general, the partial derivative of an n-ary function $f(x_1, ..., x_n)$ in the direction xi at the point $(a_1, ..., a_n)$ is defined to be [6]:

$$\frac{\partial f}{\partial x_i}(a_1, ..., a_n) = lim_{h \to 0} \frac{f(a_1, ..., a_i + h, ..., a_n) - f(a_1, ..., a_i, ..., a_n)}{h}$$

Partial derivatives can also be taken for higher orders just like ordinary derivatives.

To denote the change of f with respect to more than one variable

$$\frac{\partial^2 f}{\partial x \partial y}$$

also known as a mixed partial derivative.

2 Hessian

2.1 Definition of Gradient

Suppose that $f : R^{m \times n} x \mapsto R$ is a function that takes a matrix A of size mn and returns a real value. Then the gradient of f (with respect to $A \in R^{m \times n}$) is the matrix of partial derivatives, defined as [5]:

$$\nabla_A f(A) \in R^{m \times n} = \begin{bmatrix} \frac{\partial f(A)}{\partial A_{11}} & \frac{\partial f(A)}{\partial A_{12}} & \cdots & \frac{\partial f(A)}{\partial A_1} \\ \frac{\partial f(A)}{\partial A_{21}} & \frac{\partial f(A)}{\partial A_{22}} & \cdots & \frac{\partial f(A)}{\partial A_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f(A)}{\partial A_{m1}} & \frac{\partial f(A)}{\partial A_{m2}} & \cdots & \frac{\partial f(A)}{\partial A_{mn}} \end{bmatrix}$$

The gradient only exists for scalar valued functions (as it maps to R^1). Analogous to the gradient for scalar valued functions is the Jacobian matrix that maps vector valued functions $R^n \to R^m$, when $m = 1$ the Jacobian and the gradient is the same. Similarly to how derivatives of single variable functions are the rate of change, the gradient can also be seen as a rate of change.

2.2 Definition of Transpose

The transpose of a matrix A, denoted A^t, is the matrix obtained from A by interchanging the rows and columns. More specifically, if A is an m-by-n matrix, then A^t is the n-by-m matrix whose entries are given by the equation [1]

$$(A^t)_{k,j} = A_{j,k}$$

.

2.3 Definition of the Hessian

Suppose $f : R^n \to R^m$ is a function taking a vector $x \in^n$ as an input and outputting a scalar $f(x) \in R$; if all second partial derivatives of f exist and are continuous over the domain of the function, then the Hessian matrix with respect to x is a square $n \times n$ matrix, usually defined and arranged as follows [5]:

$$\nabla_x^2 f(x) \in R^{m \times n} = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ \cdots & \cdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix}$$

Example of Hessian

$$(f)_{ij} \equiv \frac{\partial^2 f}{\partial x_i \partial x_j} \tag{1}$$

$$\left(\frac{x^2}{y}\right) = \begin{bmatrix} \frac{2}{y} & -\frac{2x}{y^2} \\ -\frac{2x}{y^2} & \frac{2x^2}{y^3} \end{bmatrix} \tag{2}$$

2.4 Symmetry of the Hessian

The Schwarz Theorem says if partial derivatives

$$\frac{\partial^2 f}{\partial x_i \partial x_j}$$

and

$$\frac{\partial^2 f}{\partial x_j \partial x_i}$$

of function f are continuous at x_0, then they are equal. [7] The partial derivatives will be commutative. Thus the Hessian will have a symmetric matrix under these conditions. See citation for complete proof of Schwarz Thereom.

3 Gradient Descent

3.1 Cauchy Schwarz Inequality

Suppose u, v element of V. Then,

$$|\langle u, v \rangle| \leq \|u\| \, \|v\|$$

. If and only if u, v is a scalar multiple of the other. [1] The full proof is shown as 6.15 in Axler.

$$\langle \nabla f(x), d \rangle \, , \|d\| = 1$$

is the rate of increase of f at point x in direction d. Following the Cauchy Schwarz Inequality it follows that for unit vector

$$\|d\| = 1$$

$$\langle \nabla f(x), d) \rangle \leq \|\nabla f(x)\|$$

.

For

$$d = \nabla f(x) / \, \|\nabla f(x)\|$$

,

then

$$\left\langle \nabla f(x), \frac{\nabla f(x)}{\|\nabla f(x)\|} \right\rangle = \|\nabla f(x)\|$$

. [2]

This points towards the maximum rate of increase of f at x. The negative of the gradient f(x) is the maximum rate of decrease. This is useful in optimizing for a minimum.

3.2 First-Order Necessary Condition (FONC)

Let fi be a subset of R^n and f is an element of C^1 a real valued function of fi. If x is a local miner of f over fi then for any feasible direction d at x: [2]

$$d^T \nabla f(x^*) \geq 0$$

Proof: Define

$$x(\alpha) = x^* + \alpha d \ \in \ \Omega$$

$$x(0) = x^*$$

Define the composite function:

$$\phi(a) = f(x(a))$$

Then applying Taylor's theorem:

$$f(x^* + \alpha d) - f(x^*) = \phi(\alpha) - \phi(0) = \phi(0)\alpha + o(\alpha) = \alpha d^T \nabla f(x(0)) + o(\alpha)$$

For small values of $\alpha \bar{} 0$ so $x*$ is a local minimizer then there must be $d^T \nabla f(x^*) \geq 0$.

3.3 Propositions

If $x^{(k)}{}_{k=0}^{\infty}$ is a steepest descent sequence for a given function $f : R^n \to R$, then for each k the vector $x^{(k+1)} - x^{(k)}$ is orthogonal to the vector $x^{(k+2)} - x^{(k+1)}$ [2]. Proof [2]:

From the iterative formula of the method of the steepest descent it follows that

$$\left\langle x^{(k+1)} - x^{(k)}, x^{(k+2)} - x^{(k+1)} \right\rangle = \alpha_k \alpha_{k+1} \left\langle \nabla f(x^{(k)}), \nabla f(x^{(k+1)}) \right\rangle$$

$$\nabla f(x^{(k)}), \nabla f(x^{(k+1)}) = 0$$

α_k is a non negative scalar that minimizes $\phi_k(\alpha)$ defined as $f(x^{(k)} - \alpha \nabla f(x^{(k)})$. Using FONC and the chain rule,

$$0 = \phi_k'(\alpha_k) = \frac{d\phi_k}{d\alpha}(\alpha_k) = \nabla f(x^{(k)} - a_k \nabla f(x^{(k)})^T (-\nabla f(x^{(k)})) = -\left\langle \nabla f(x^{(k+1)}), \nabla f(x^{(k)}) \right\rangle$$

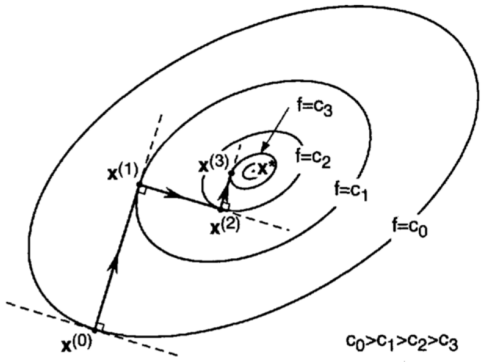
This says that $\nabla f(x^(k)$ is parallel to the tangent plane to the level set $f(x) = f(x^{(k+1)})$ at $x^{(k+1)}$.

If $x^{(k)}{}_{k=0}^{\infty}$ is the steepest descent sequence for $f : R^n \to R$ and if $\nabla f(x^{(k)}) \neq 0$ then $f(x^{(k+1)}) f(x^{(k)})$ [2] (proof follows in citation).

Thus, the steepest descent algorithm must contain the property of descent and these two propositions provide the foundational criteria for which the algorithm will terminate.

3.4 Steepest Descent Algorithm

Steepest descent is used to find the maximum decrease between individual steps for a function. Starting from a fixed point x^k, the direction of the negative gradient $-\nabla f(x^K)$ that we saw in the Cauchy Schwarz inequality is used to find a minimizer, $x^(k+1)$. See figure below for illustration.



This looks like:

$$a_k = argmin_{(a\geq 0)} f(x^k) - a\nabla f(x^k)$$

Proof follows [2]:

$$\phi_k(\alpha) = f(x^{(k)} - \alpha(x^{(k)}$$

$$\phi_k(\alpha_k) \leq \phi_k(a)$$

$$\phi_k = \frac{d\phi_k}{d\alpha}(0) = -(\nabla f(x^{(k)} - 0\nabla f(x^{(k)}) = -\left\|\nabla f(x^{(k)})\right\|^2 < 0$$

$$f(x^{(k+1))}) = \phi_k(a_k) \leq \phi_k(\bar{\alpha}) < \phi(0) = f(x^k)$$

Here *argmin* refers to the function locating the minimum value. The algorithm uses the gradient as defined in the Cauchy Schwarz Inequality. This provides the condition to terminate once the minimizer is found.

For quadratic formulas, the hessian and the property of second derivative symmetry (Schwarz Thereom) (or the assumption of symmetry since this is a quadratic) is used for steepest descent:

With a quadratic of the form:

$$f(x) = \frac{1}{2}x^T Qx - b^T x$$

Q is a symmetric square positive definite matrix, $b, x \in R^n$ so the gradient of f(x) is equal to $Qx - b$ following $D(x^T Qx) = x^T(Q+Q)^T) = 2x^T Q and D(b^T x) = b^T$. Since it is a square matrix, generality is not lost [2]. Thus using properties of transpose:

$$(x^T Ax)^T = x^T A^T x = x^T Ax$$

$$x^T Ax = \frac{1}{2}x^T Ax + \frac{1}{2}x^T A^T x$$

$$\frac{1}{2}x^T(A + A^T)x$$

is defined to be:

$$\frac{1}{2}x^T Qx$$

The Hessian of $f = Q = Q^T > 0$ (symmetrical). The steepest descent algorithm for the quadratic function is represented as

$$x^{(k+1)} = x^{(k)} - a_k g^{(k)},$$

where

$$a_k = argmin_{\alpha \geq 0} f(x^{(k)} - \alpha g^{(k)})$$

$$= argmin_{\alpha \geq 0}(\frac{1}{2}(x^{(k)} - \alpha g^{(k)})^T Q(x^{(k)} - \alpha g^{(k)}) - (x^{(k)} - \alpha g^{(k)})^T b)$$

Assuming $g^(k)$ does not equal 0 (otherwise it stops), using FONC we get:

$$\phi'_k(\alpha) = (x^{(k)} - \alpha g^{(k)})^T Q(-g^{(k)}) - b^T(-g^{(k)})$$

Therefore,

$$\phi'_k(\alpha) = 0 \, if \, \alpha g^{(k)T} Q g^{(k)} = (x^{(k)T} Q - b^T) g^{(k)}. \, But \, x^{(k)T} Q - b^T = g^{(k)T}.$$

Hence,

$$a_k = \frac{g^{(k)T} g^{(k)}}{g^{(k)T} Q g^{(k)}}$$

4 Applications

The gradient and Hessian are integral parts of the steepest decline algorithm. This algorithm has applications in feedforward neural networks. Information flows in one direction through these networks and approximate some function f and learns parameters to create the best approximation [8]. Neural networks are important in machine learning and are modeled after human brains. One real world example of feedforward neural networks is self driving cars [4].

References

[1] Axler, Sheldon. "Linear Algebra Done Right." Undergraduate Texts in Mathematics.

[2] Choong, Edwin. "An Introduction to Optimization." Wiley-Interscience Series in Discrete Mathematics and Optimization.

[3] Dawkins, Paul. "Section 2-2 : Partial Derivatives." From Paul's Online Notes. http://tutorial.math.lamar.edu/Classes/CalcIII/PartialDerivatives.aspx

[4] Di Palo, Norman. "How to Train your Self-Driving Car to Steer " From Towards Data Science. https://towardsdatascience.com/how-to-train-your-self-driving-car-to-steer-68c3d24bbcb7

[5] Kolter, Zico. "Linear Algebra Review and Reference." From Carneige Melon University School of Computer Science. http://www.cs.cmu.edu/ zkolter/course/15-884/linalg-review.pdf

[6] Weisstein, Eric W. "Partial Derivative." From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/PartialDerivative.html

[7] Cwikel, Michael. "SCHWARZ' THEOREM ABOUT MIXED PARTIAL DERIVATIVES, WITH SOME PRELIMINARY COMMENTS AND SOME REMARKS ABOUT WORKING WITH DERIVATIVES ON A COMPUTER." From Department of Mathematics Insuitite of Technology Israel. http://www2.math.technion.ac.il/ mcwikel/h2m/SchwarzFxyFyx.pdf

[8] Tushar, Gupta. "Deep Learning: Feedforward Neural Network" From Towards Data Science. https://towardsdatascience.com/deep-learning-feedforward-neural-network-26a6705dbdc7