Gradient Optimization Q&A

Kevin Qi

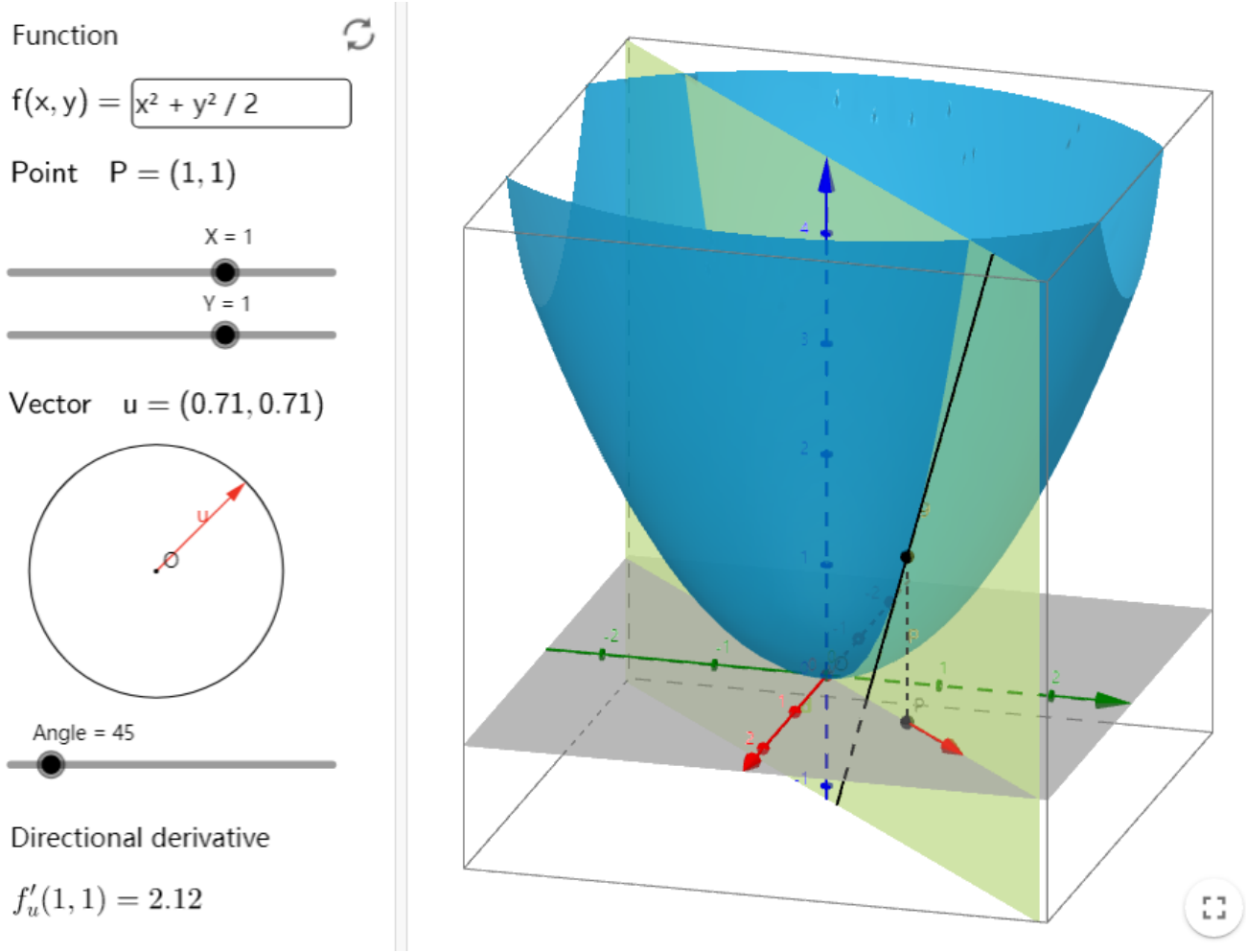
Q1 (4.3)

For Gradient Optimization, it shows us the method of steepest descent and directional derivative. And I am curious about the relationship between these two.

A1

The method of steepest descent (gradient descent) uses gradient to iteratively update the input. So I'm going to explain the difference between **gradient** and **directional derivative**.

Take a binary function $f(x, y) = (x^2 + y^2)/2$ as example.



(Figure from geogebra.org/m/Bx8nFMNc)
Gradient, as we all know, is a vector

$$\nabla f = (\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}).$$

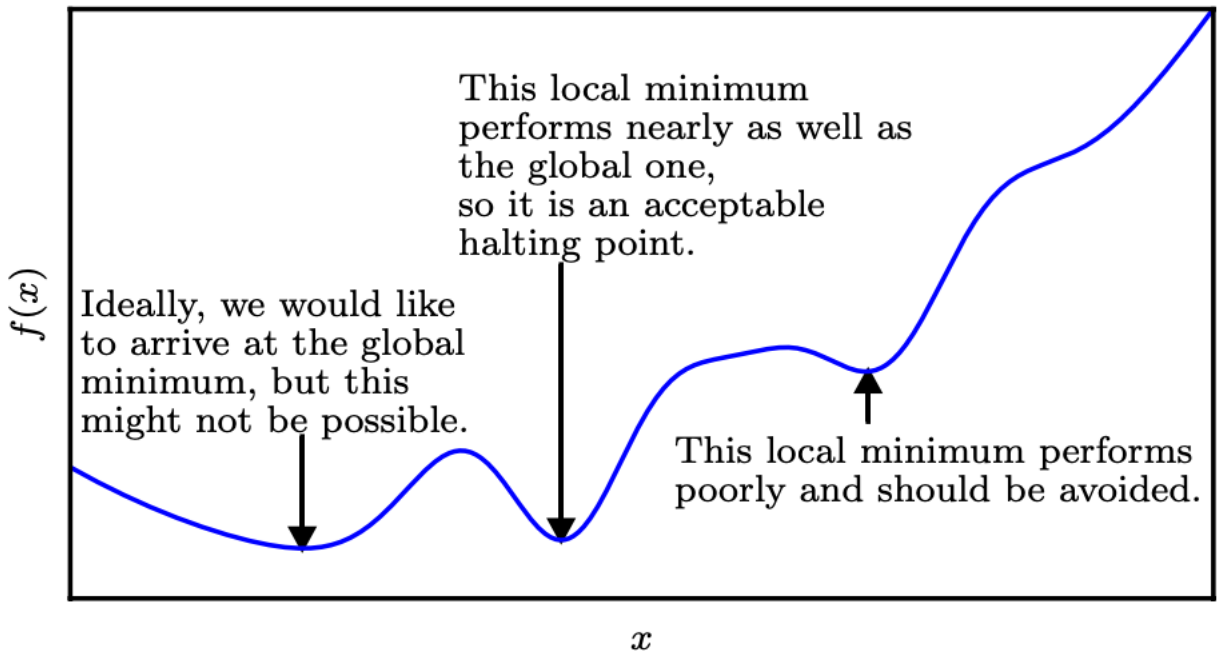
By contrast, directional derivative is a scalar (number). It's just like if we are climbing a mountain in a direction(meridian and parallel as x and y axis), how steep the mountain would be along this certain direction. The directional derivative of function $f(x, y)$ along vector $\vec{v}$, we just have

$$\nabla_v f = \nabla f \cdot \vec{v}.$$

The partial derivatives are examples of directional derivatives. $\frac{\partial f}{\partial x}$ along $\vec{e}_x = (1, 0)$.

Q2 (4.3)

For Gradient Opt, in figure 4.3,



the right most local minimum should be avoided, but I'm wondering how that local minimum can be avoided? Didn't found answer in the rest of the chapter.

A2

If we simply use a learning rate multiplies the gradient to update the input, it would be easily stuck in a local minimum. There is a large amount of

algorithms trying to fix it.

- Momentum
  https://ruder.io/optimizing-gradient-descent/index.html#momentum
  The algorithm optimizes by adding a fraction γ of the update vector of the past time step to the current update vector:

$$v_t = \gamma v_{t-1} + \eta \nabla_\theta J(\theta),$$
$$\theta = \theta - v_t.$$

  This method applies the idea of inertia and friction.
  Example: a ball rolls down from somewhere.

- Simulated annealing (SA)
  https://en.wikipedia.org/wiki/Simulated_annealing

Q3 (4.3.1)

In gradient optimization reading, what to do with example as figure 4.5? Can we still find maximum value in this case?

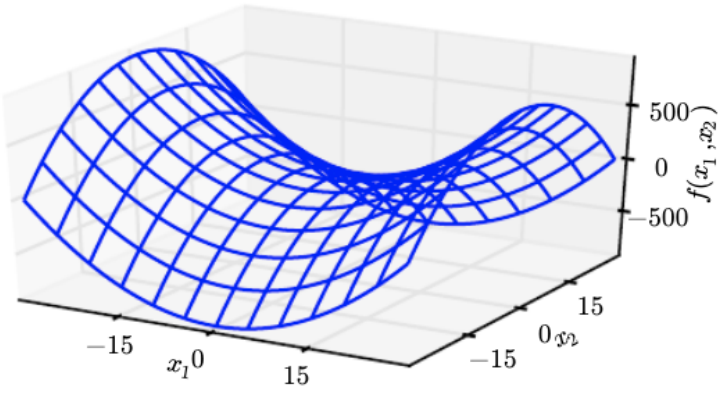


Figure 4.5: A saddle point containing both positive and negative curvature. The function in this example is $f(\boldsymbol{x}) = x_1^2 - x_2^2$. Along the axis corresponding to x_1, the function curves upward. This axis is an eigenvector of the Hessian and has a positive eigenvalue. Along the axis corresponding to x_2, the function curves downward. This direction is an eigenvector of the Hessian with negative eigenvalue. The name "saddle point" derives from the saddle-like shape of this function. This is the quintessential example of a function with a saddle point. In more than one dimension, it is not necessary to have an eigenvalue of 0 to get a saddle point: it is only necessary to have both positive and negative eigenvalues. We can think of a saddle point with both signs of eigenvalues as being a local maximum within one cross section and a local minimum within another cross section.

A3

If we use first-order optimization algorithm like gradient descend, we would be stuck in point $(0, 0)$. However, we can use Hessian matrix (based on second derivatives) to get out. The Hessian matrix of $f(\boldsymbol{x}) = x_1^2 - x_2^2$ is:

$$\boldsymbol{H}(f)(\boldsymbol{x}) = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \qquad \text{for every } \boldsymbol{x}$$

The eigenvalues of this matrix are $\lambda_1 = 1, \lambda_2 = -1$.

Each element of the Hessian matrix is constant, and its two λs have different signs, so in this case, the function has neither a global minimum nor maximum.

Q4

According to the textbook:

> The optimization algorithms employed in most contexts in this book are applicable to a wide variety of functions

So there should be the cases where there are not much saddle points in the loss function that we can use second-order optimization algorithms, such as Newton's method. But it seems that people barely use second-order optimization algorithms, is it because nowadays loss function always contains much saddle points? Or is there some other reasons that are blocking the use of Newton's method?

A4

- https://stats.stackexchange.com/questions/253632/why-is-newtons-method-not-widely-used-in-machine-learning
- https://zhuanlan.zhihu.com/p/110922480

Second-order optimization algorithms are way more expensive. That can be faster when the second derivative is known and easy to compute, but the analytic expression for the second derivative is often complicated or

intractable, requiring a lot of computation.

$$
\begin{aligned}
\frac{\partial^2 E}{\partial w_{ji}^{(1)} \partial w_{j'i'}^{(1)}} &= \frac{\partial}{\partial w_{ji}^{(1)}} \left(\sum_{k=1}^{K} \frac{\partial E}{\partial y_k} w_{kj'}^{(2)} h'(a_{j'}) x_{i'} \right) \\
&= \sum_{k=1}^{K} \frac{\partial}{\partial w_{ji}^{(1)}} \left(\frac{\partial E}{\partial y_k} h'(a_{j'}) \right) w_{kj'}^{(2)} x_{i'} \\
&= \sum_{k=1}^{K} \frac{\partial}{\partial w_{ji}^{(1)}} \left(\frac{\partial E}{\partial y_k} h'(a_j) \right) w_{kj}^{(2)} x_{i'} \\
&= \sum_{k=1}^{K} \left(\frac{\partial}{\partial w_{ji}^{(1)}} \left(\frac{\partial E}{\partial y_k} \right) h'(a_j) + \frac{\partial h'(a_j)}{\partial w_{ji}^{(1)}} \frac{\partial E}{\partial y_k} \right) w_{kj}^{(2)} x_{i'} \\
&= \left(\sum_{k=1}^{K} \sum_{k=1}^{K'} \frac{\partial^2 E}{\partial y_k y_{k'}} w_{k'j}^{(2)} h'(a_j) h'(a_j) x_i + \sum_{k=1}^{K} \frac{\partial h'(a_j)}{\partial a_j} \frac{\partial a_j}{\partial w_{ji}^{(1)}} \frac{\partial E}{\partial y_k} \right) w_{kj}^{(2)} x_{i'} \\
&= x_{i'} x_i h'(a_j) h'(a_j) \sum_{k=1}^{K} \sum_{k=1}^{K'} \frac{\partial^2 E}{\partial y_k y_{k'}} w_{k'j}^{(2)} w_{kj}^{(2)} + \sum_{k=1}^{K} h''(a_j) x_i \frac{\partial E}{\partial y_k} w_{kj}^{(2)} x_{i'} \\
&= x_{i'} x_i (h'(a_j))^2 \sum_{k=1}^{K} \sum_{k=1}^{K'} M_{kk'} w_{k'j}^{(2)} w_{kj}^{(2)} + x_{i'} x_i h''(a_j) \sum_{k=1}^{K} \delta_k w_{kj}^{(2)}
\end{aligned}
$$

Q5-1 (4.3.1~4.4)

On gradient descent, there are multiple ways to give 'guarantees'. The Lipschitz, convex optimization, and constrained optimization methods.

Wouldn't using the lipschitz or convex optimization still give optimal results?

A5-1

- Lipschitz optimization

$$\forall \boldsymbol{x}, \forall \boldsymbol{y}, |f(\boldsymbol{x}) - f(\boldsymbol{y})| \leq \mathcal{L} ||\boldsymbol{x} - \boldsymbol{y}||_2.$$

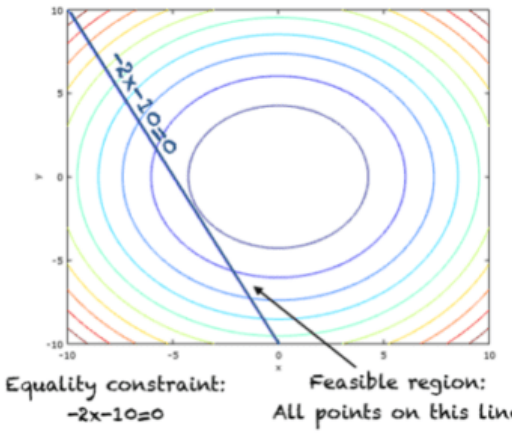
  -
  - Minimal step, underflow & rounding error
    - Acceptable error: $L \cdot minimal\_step$, or $O(minimal\_step)$
    - Without the Lipschitz continuous
- Convex optimization
  - Can be easily done by gradient descent
  - Counter-example: villages in a mountainous area

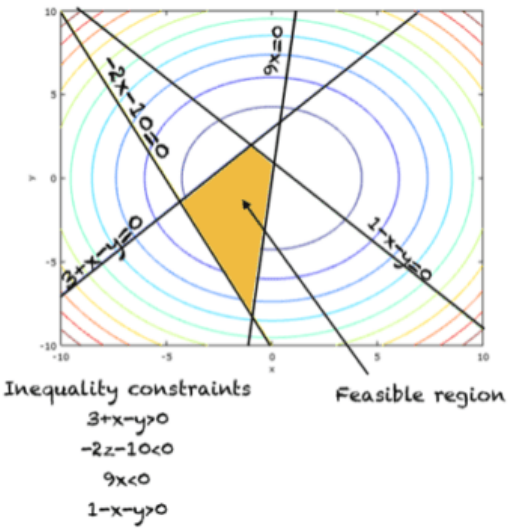
Q5-2(4.4)

Also how do you choose the 'feasible points' during constrained optimization?

A5-2

- Equality Vs. Inequality Constraints





- Equality Constraints -- Lagrange multiplier
  - in order to find the maximum or minimum of a function $f(x)$ subjected to the equality constraint $g(x) = 0$, form the **Lagrangian function**

$$L(x, \lambda) = f(x) + \lambda g(x).$$

    and find the stationary pointsof L considered as a function of x and the Lagrange multiplier λ .
- Inequality Constraints -- KKT Method (on the textbook)

Q6-1 (4.5)

This is a mathematical question. In gradient optimization reading 4.5 Example, how do we calculate gradient in 4.22? why

$\nabla_x f(x) = A^T(Ax - b)$?

4.5 Example: Linear Least Squares

Suppose we want to find the value of $\boldsymbol{x}$ that minimizes

$$f(\boldsymbol{x}) = \frac{1}{2}||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}||_2^2. \tag{4.21}$$

Specialized linear algebra algorithms can solve this problem efficiently; however, we can also explore how to solve it using gradient-based optimization as a simple example of how these techniques work.

First, we need to obtain the gradient:

$$\nabla_{\boldsymbol{x}} f(\boldsymbol{x}) = \boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) = \boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}^\top\boldsymbol{b}. \tag{4.22}$$

A6-1 (4.5)

https://en.wikipedia.org/wiki/Matrix_calculus

Let's have a look at (4.21).

$$\nabla_{\boldsymbol{x}} f(x) = \frac{1}{2}||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}||_2^2$$

First we have

$$\nabla_{\boldsymbol{x}} f(\boldsymbol{x}) = \left(\frac{df}{d\boldsymbol{x}}\right)^T = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$$

and

$$||\boldsymbol{x}||_2 = \sqrt{\sum_i x_i^2},$$

$$\frac{1}{2}||\boldsymbol{x}||_2^2 = \frac{1}{2}\sum_i x_i^2.$$

Therefore,

$$\frac{d(\frac{1}{2}||\boldsymbol{x}||_2^2)}{dx_i} = \frac{d(\frac{1}{2}x_i^2)}{dx_i} = x_i,$$

$$\frac{d(\frac{1}{2}||\boldsymbol{x}||_2^2)}{d\boldsymbol{x}} = \boldsymbol{x}^T.$$

Here the left side is a row column, right side
Also we have

$$\frac{d(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b})}{d\boldsymbol{x}}=\boldsymbol{A}$$

According to the chain rule,

$$\frac{df}{d\boldsymbol{x}}=\frac{d(\frac{1}{2}||\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}||_2^2)}{d(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b})}\cdot\frac{d(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b})}{d\boldsymbol{x}}=(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b})^T\cdot\boldsymbol{A}$$

$$\nabla_{\boldsymbol{x}}f(\boldsymbol{x})=\left(\frac{df}{d\boldsymbol{x}}\right)^T=\boldsymbol{A}^T\cdot(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b})$$

Notice that $\boldsymbol{x}$ is a column vector, but a scale-by-vector direvative $\frac{df}{d\boldsymbol{x}}$ is a row vector.

Q6-2

And how do we get 4.25 from 4.23 by differentiating the Lagrangian with respect to x? I feel like these two questions involve similar formula of gradient of a vector, so I asked these two together...

$$L(\boldsymbol{x}, \lambda) = f(\boldsymbol{x}) + \lambda \left(\boldsymbol{x}^\top \boldsymbol{x} - 1 \right). \tag{4.23}$$

We can now solve the problem

$$\min_{\boldsymbol{x}} \max_{\lambda, \lambda \geq 0} L(\boldsymbol{x}, \lambda). \tag{4.24}$$

The smallest-norm solution to the unconstrained least-squares problem may be found using the Moore-Penrose pseudoinverse: $\boldsymbol{x} = \boldsymbol{A}^+ \boldsymbol{b}$. If this point is feasible, then it is the solution to the constrained problem. Otherwise, we must find a

solution where the constraint is active. By differentiating the Lagrangian with respect to $\boldsymbol{x}$, we obtain the equation

$$\boldsymbol{A}^\top \boldsymbol{A} \boldsymbol{x} - \boldsymbol{A}^\top \boldsymbol{b} + 2\lambda \boldsymbol{x} = 0. \tag{4.25}$$

A6-2

Yeah it's basically the same question. For a column vector $\boldsymbol{x}$,

$$\boldsymbol{x}^T \boldsymbol{x} = ||\boldsymbol{x}||_2^2.$$