Chapter 4. Numerical Computation

4.1 Underflow and Overflow

Very small value will be stored as zero in computer

4.2 Poor condition

- Conditioning refers to how rapidly a function changes with respect to small changes in its inputs
- High condition number suffer more from underflow and overflow

Condition number

For a $n \times n$ matrix A, if A has an eigenvalue decomposition with eigenvalue $\{\lambda_i, \lambda_2, \dots, \lambda_n\}$, its condition number is

$$\max_{i,j} |\frac{\lambda_i}{\lambda_j}|$$

4.3 Gradient-Based Optimization

Overview

Gradient Descent:

Reduce $f(x)$ by move x in small step with opposite sign of its derivative

For multivariable function $f(\mathbf{x})$, there are many direction that can reduce $f(\mathbf{x})$, but the steepest descent direction is

$\mathbf{x}' = \mathbf{x} - \epsilon \nabla_x f(\mathbf{x})$

ϵ is the learning rate

How to optimize learning rate ϵ

- Line search: given a range $[a, b]$, search the optimal ϵ by increasing a step size every time

4.3.1 Jacobian and Hessian Matrixes

Jacobian:

Here is a really good post for Jacobian

Definition: For a function $f : R^m \to R^n, J \in R^{n \times m}$ such that.

$$J_{i,j} = \frac{\partial f_i}{\partial x_j}$$

Hessian:

For a function $f : R^n \to R, H \in R^{n \times n}$

$$H_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}$$

Relationship between Jacobian and Hessian

For a function $f : R^n \to R$, the gradient of f is a function $\nabla f : R^n \to R^n$

$$\nabla f(\mathrm{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix}$$

Then the Jacobian of this gradient ∇f is the Hessian of f

$$H(\mathrm{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^n f}{\partial x_1 \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{bmatrix}$$

Find optimal learning rate with Hessian matrix

Find the second-order Taylor approximation to a function $f(\mathrm{x}) : R^n \to R$ around the point $x^{(0)}$.

$$f(x) = f(x^{(0)}) + (x - x^{(0)})^\top g + \frac{1}{2}(x - x^{(0)})^\top H(x - x^{(0)})$$

Where g is the gradient and H is the Hessian. Since the gradient descent usually take a relatively small step around $x^{(0)}$, then the value after gradient descent step is

$$f(x^{(0)} - \epsilon g) = f(x^{(0)}) - \epsilon g \top g + \frac{1}{2}\epsilon^2 g \top H g$$

Our objective is usually minimize $f(x)$ by gradient descent, then

- If H is negative semidefinite, the optimal ϵ is positive infinite
- if H is positive definite, the optimal ϵ is

$$\epsilon = \frac{g \top g}{g \top H g}$$

First order and second order optimization

- First order test can find the critical points of a function f
- Second order test can determine whether a critical point is local minimum local maximum or saddle point.
- For a critical point of a function $f : R^n \to R$
  - If Hessian H of f is positive definite, the critical point is a local minimum
  - If Hessian H of f is negative definite, the critical point is a local maximum

Guide gradient descent with Hessian information

If the Hessian matrix H has poor condition number, gradient descent performs poorly.

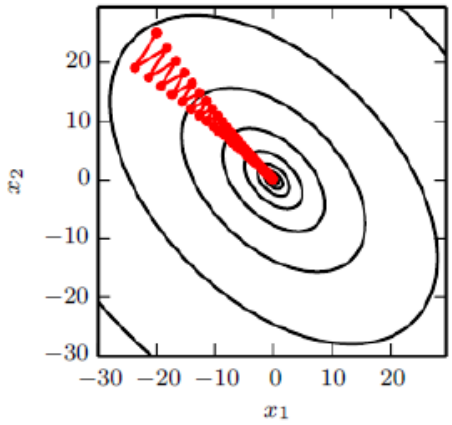

Figure 4.6: Gradient descent fails to exploit the curvature information contained in the Hessian matrix. Here we use gradient descent to minimize a quadratic function $f(\boldsymbol{x})$ whose Hessian matrix has condition number 5. This means that the direction of most curvature has five times more curvature than the direction of least curvature. In this case, the most curvature is in the direction $[1, 1]^\top$ and the least curvature is in the direction $[1, -1]^\top$. The red lines indicate the path followed by gradient descent. This very elongated quadratic function resembles a long canyon. Gradient descent wastes time repeatedly descending canyon walls, because they are the steepest feature. Because the step size is somewhat too large, it has a tendency to overshoot the bottom of the function and thus needs to descend the opposite canyon wall on the next iteration. The large positive eigenvalue of the Hessian corresponding to the eigenvector pointed in this direction indicates that this directional derivative is rapidly increasing, so an optimization algorithm based on the Hessian could predict that the steepest direction is not actually a promising search direction in this context.

下降最快未必是最好的

The solution is use Hessian information to guide the gradient descent. The simplest method is use **Newton's Method**. Firstly, find the second order Taylor approximation of $f(x)$ at target point $x^{(0)}$

$$f(x) = f(x^{(0)}) + (x - x^{(0)})^\top g + \frac{1}{2}(x - x^{(0)})^\top H(x - x^{(0)})$$

Then we optimize this second order approximation and get the *critical point*

$$x^\star = x^{(0)} - H(f)(x^{(0)})^{-1}\nabla_x f(x^{(0)})$$

NOTE:

- If near a local minimum, Newton's method perform much better than gradient descent without Hessian information
- If near saddle point, Newton's method is harmful

Lipschitz continuous

Used frequently in non-convex optimization

$$\forall x, \forall y, |f(x) - f(y)| \leq L||x - y||_2$$

4.4 Constrained Optimization

Lagrangian multiplier and KKT approach

Convert constrained optimization to a set of linear equation system

4.5 Example: Linear Least Square

Refer to the book