Class #10

Friday, February 21, 2025 10:18 AM

Chain rule for Hessians

① Suppose $f: \mathbb{R}^n \to \mathbb{R}$, $g: \mathbb{R} \to \mathbb{R}$

$$h(x) = g(f(x))$$

$$\nabla^2 h(x) = g'(f(x))\, \nabla^2 f(x) + g''(f(x))\, \nabla f(x) \nabla f(x)^T$$

② Suppose $f: \mathbb{R}^n \to \mathbb{R}$

$g: \mathbb{R}^m \to \mathbb{R}$

$A \in \mathbb{R}^{n \times m}$

$b \in \mathbb{R}^n$

$$g(x) = f(Ax+b)$$

$$\nabla^2 g(x) = A^T \nabla^2 f(Ax+b) A$$

③ Define $\tilde{f}(t) = f(x+tv)$

$$\nabla^2 \tilde{f}(t) = \tilde{f}''(t) = v^T \nabla^2 f(x+tv) v$$

For $t=0$: $\nabla^2 \tilde{f}(0) = v^T \nabla^2 f(x) v$

Example: $f(x) = \frac{1}{2} x^T P x + q^T x + r$

$P \in S^n$, $q \in \mathbb{R}^n$, $r \in \mathbb{R}$

$$\nabla f(x) = Px + q$$

$$\nabla^2 f(x) = D(\nabla f(x)) = P$$

assume: $f(x) = \frac{1}{2} \boxed{a} x^2 + bx + c$

$$f''(x) = a$$

Newton's Method

It is "supposed" to be a descent method with iterations given by:

$$x^{(k+1)} = x^{(k)} + t^{(k)} \Delta x_{nt}$$

where: $\Delta x_{nt} = -\left[\nabla^2 f(x)\right]^{-1} \nabla f(x)$ Newton direction

$\Downarrow$

Based on this, it requires:

1. Function f has to be twice differentiable
   - Typically we assume $f \in C^2$
   - Twice continuously differentiable
2. $\nabla^2 f(x)$ must be invertible $\Rightarrow \text{rank}(\nabla^2 f(x)) = n$
   - Typical requirement is that it is invertible over every $x \in \mathbb{R}^n$
3. In order for Newton's method to be a descent method, we require that

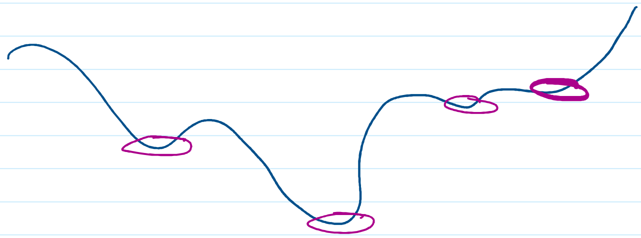
$$\nabla^2 f(x) \succ 0 \quad \text{(Positive definite)}$$

(Remember: $A \succ 0 \Leftrightarrow A^{-1} \succ 0$)

There are two ways to handle ③

1. Assume $\nabla^2 f(x) \succ 0 \quad \forall x \in \mathbb{R}^n$
   - strongly convex functions

② What about non convex functions?



In that case, we first run gradient descent for a number of iterations till $\|\nabla f(x)\|_2$ is small and then we switch to Newton iterations.

Even when $\nabla^2 f(x) \succ 0 \ \forall x \in \mathbb{R}^n$, Newton's has some drawbacks:

① Compute and store $\nabla^2 f(x)$

② Compute inverse of $[\nabla^2 f(x)]^{-1}$

So why use it? It is extremely fast in the right regions (to be shown later).

Ways to deal with these issues

① Quasi-Newton method

② Approximate the Hessian by looking/exploiting the structure of the problem (in a fast way).

E.g.,
$$\nabla f(x) = \begin{bmatrix} f_1(x_1) \\ f_2(x_2) \\ \vdots \\ f_n(x_n) \end{bmatrix}$$

$$\Rightarrow \nabla^2 f(x) = \begin{bmatrix} f_1'(x_1) & & & 0 \\ & f_2'(x_2) & & \\ & & \ddots & \\ 0 & & & f_n'(x_n) \end{bmatrix}$$

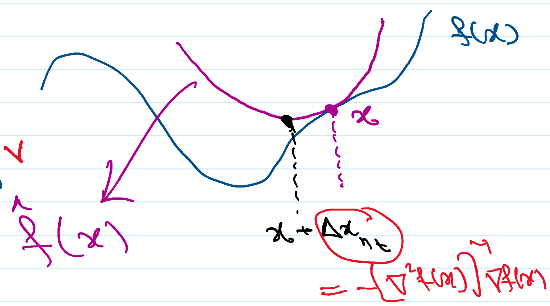
Interpretations of Newton's Method

① Minimizer of the second-order approximation of f at x

① Minimizer of the second-order approximation of f at x

$$\hat{f}(y) = f(x) + \nabla f(x)^T (y-x) + \frac{1}{2}(y-x)^T \nabla^2 f(x)(y-x)$$

write $y = x + v$



$$\hat{f}(x+v) = \underbrace{f(x) + \nabla f(x)^T v + \frac{1}{2} v^T \nabla^2 f(x) v}$$

argmin

$$= -[\nabla^2 f(x)]^{-1} \nabla f(x)$$

Compute ∇_v and set it equal to 0.

$$\nabla_v \left(f(x) + \nabla f(x)^T v + \underbrace{\frac{1}{2} v^T \nabla^2 f(x) v} \right)$$

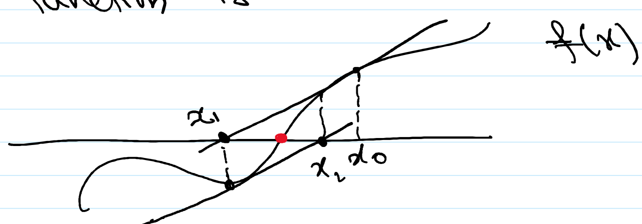
$$= 0 + \nabla f(x) + \nabla^2 f(x) v = 0$$

$$\nabla^2 f(x) v = -\nabla f(x)$$

$$v^* = -[\nabla^2 f(x)]^{-1} \nabla f(x)$$

② Newton's method is also tied to the idea of approximating the gradient $\nabla f(x)$ by a linear function and then finding the root of that linear function.

Stationary point of a function is when $\nabla f(x) = 0$



Given $f(x) : \mathbb{R} \to \mathbb{R}$

$$\underbrace{f(y)}_{0} \approx f(x) + f'(x)(y-x)$$

$$\nabla f(y) \sim \nabla f(x) + \nabla^2 f(x)(y-x)$$

$$\nabla f(y) \approx \nabla f(x) + \nabla^2 f(x)(y-x)$$

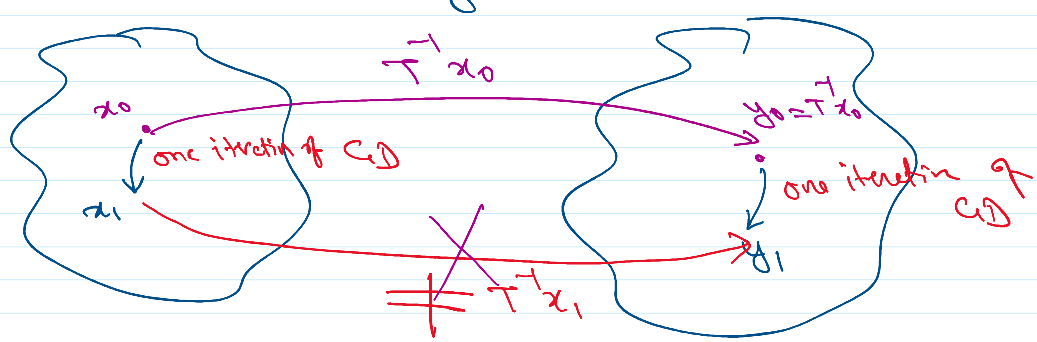
Put $y = x + v$

$$\underbrace{\nabla f(x+v)}_{=0} = \nabla f(x) + \nabla^2 f(x)\, v$$

$$\nabla^2 f(x)\, v = -\nabla f(x)$$

$$v = -\left[\nabla^2 f(x)\right]^{-1} \nabla f(x)$$

$$x \in \mathbb{R}^n \underset{Ty}{\overset{T^{-1}x}{\rightleftarrows}} y \in \mathbb{R}^n \qquad (T \in \mathbb{R}^{n\times n} \text{ is non-singular})$$



Gradient descent, in general, is <u>not</u> affine invariant. Coordinate system in gradient descent affects the algorithmic performance.

Affine invariance of Newton's Step

Suppose $T \in \mathbb{R}^{n\times n}$ is non-singular and

let $y = T^{-1}x$; $\boxed{x = Ty}$

$$f(x) : \mathbb{R}^n \to \mathbb{R}$$

$$\bar{f}(y) = f(Ty) : \mathbb{R}^n \to \mathbb{R}$$

$$\bar{f}(y) = f(Ty) : \mathbb{R}^n \to \mathbb{R}$$

Then: $x + \Delta x_{nt} = T(y + \Delta y_{nt})$

Basic Assumption

Either we are close to a local optimum or we are working with the case $\nabla^2 f(x) \succ 0 \;\; \forall\, x \in \mathbb{R}^n$.

Newton Decrement

$$\lambda(x) = \left[\nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x)\right]^{1/2}$$

is called Newton decrement.

1. Used in analysis
2. Used in stopping criterion of Newton's method

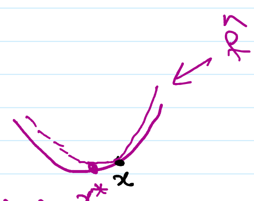
- $\lambda(x)$ is a scalar
- $\lambda(x) > 0$ since $\nabla^2 f(x)^{-1} \succ 0$

$\lambda(x)$ allows us to approximate how close we are to a local minimum (p^*).

Recall: $\hat{f}(x+v) = f(x) + \nabla f(x)^T v + \frac{1}{2} v^T \nabla^2 f(x) v$

$$\underbrace{f(x)}_{\text{function at } x} - \underbrace{\hat{f}(x+v)}_{\text{Quadratic approximation at } x} = -\nabla f(x)^T v - \frac{1}{2} v^T \nabla^2 f(x) v$$

$v = -\nabla^2 f(x)^{-1} \nabla f(x)$



$$f(x) - \min_v \hat{f}(x+v) \approx -\nabla f(x)^T\left(-\nabla^2 f(x)^{-1} \nabla f(x)\right)$$

$$f(x) - \underbrace{\min_{v} f(x+v)}_{p^*} \simeq -\nabla f(x)^T\left(-\nabla^2 f(x)^{-1}\nabla f(x)\right) - \frac{1}{2}\left(-\left(\nabla^2 f(x)^{-1}\nabla f(x)\right)^T\right)\nabla^2 f(x)\left(-\nabla^2 f(x)^{-1}\nabla f(x)\right)$$

$$\boxed{f(x) - p^* \simeq \frac{\lambda(x)^2}{2}}$$

When our current iteration x is very close to x^*, $\frac{\lambda(x)^2}{2}$ gives us an estimate of how far we are from the local minimum.