Deep Learning

Numerical Optimization

2016.08.08

이우진

Unconstrained Optimization

- In unconstrained optimization, we minimize an objective function that depends on real variables, with no restrictions at all on the values of these variables. The mathematical formulation is

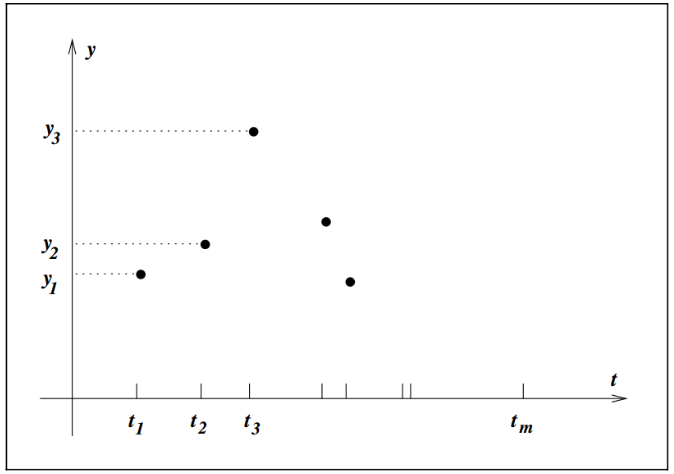
$$\min_{x} f(x),$$

- we choose to model it by the function

$$\phi(t;x) = x_1 + x_2 e^{-(x_3-t)^2/x_4} + x_5 \cos(x_6 t)$$

- Parameters : Xi , i=1,2,3,4,5,6
- Objective function $r_j(x) = y_j - \phi(t_j;x)$

$$\min_{x \in \mathbb{R}^6} f(x) = r_1^2(x) + r_2^2(x) + \cdots + r_m^2(x)$$



- 여기에서는 n=6인 문제지만 실제로는 10^5같은 문제를 풀어야 한다.

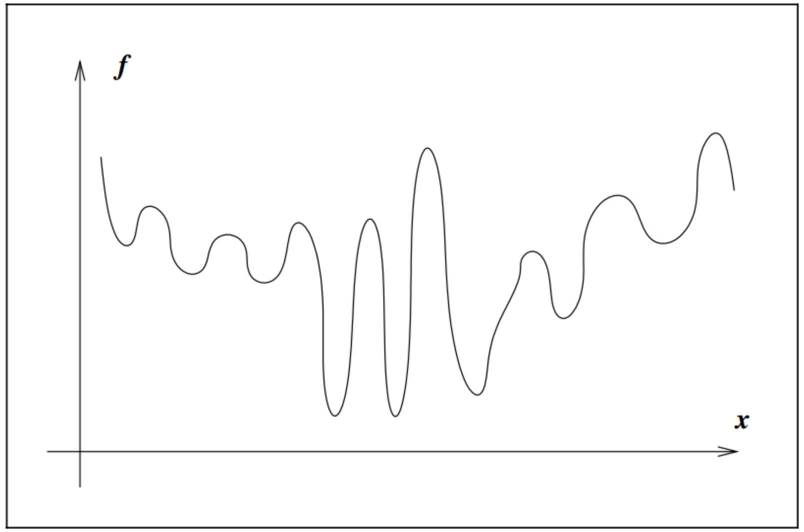
What is Solution?

- 우리가 찾아야 하는 것은 Global minimizer

  A point x^* is a *global minimizer* if $f(x^*) \leq f(x)$ for all x,

- F에 대한 전체의 그림을 알 수 없고, 알고리즘이 많은 포인트들을 찾지 않기 때문에 Global minimizer는 찾기 어렵고, local하게 solution을 찾게 된다.

  A point x^* is a *local minimizer* if there is a neighborhood $\mathcal{N}$ of x^* such that $f(x^*) \leq f(x)$ for all $x \in \mathcal{N}$.



이러한 local minimizer에 빠져버리는 알고리즘이 많다.

Recognizing a local minimum

- In particular, if f is twice continuously differentiable, we may be able to tell that x∗ is a local minimizer (and possibly a strict local minimizer) by examining just the gradient ∇ f (x∗) and the Hessian ∇^2 f (x∗).

Theorem 2.1 (Taylor's Theorem).

Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and that $p \in \mathbb{R}^n$. Then we have that

$$f(x+p) = f(x) + \nabla f(x+tp)^T p, \tag{2.4}$$

for some $t \in (0,1)$. Moreover, if f is twice continuously differentiable, we have that

$$\nabla f(x+p) = \nabla f(x) + \int_0^1 \nabla^2 f(x+tp)p\,dt, \tag{2.5}$$

and that

$$f(x+p) = f(x) + \nabla f(x)^T p + \tfrac{1}{2} p^T \nabla^2 f(x+tp)p, \tag{2.6}$$

for some $t \in (0,1)$.

Recognizing a local minimum

Theorem 2.2 (First-Order Necessary Conditions).

If x^* is a local minimizer and f is continuously differentiable in an open neighborhood of x^*, then $\nabla f(x^*) = 0$.

Theorem 2.3 (Second-Order Necessary Conditions).

If x^* is a local minimizer of f and $\nabla^2 f$ exists and is continuous in an open neighborhood of x^*, then $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive semidefinite.

- In particular, if f is twice continuously differentiable, we may be able to tell that x∗ is a local minimizer (and possibly a strict local minimizer) by examining just the gradient ∇ f (x∗) and the Hessian ∇^2 f (x∗).

Overview of Algorithms

- **Line search**
  - line search strategy, the algorithm chooses a direction pk and searches along this direction from the current iterate xk for a new iterate with a lower function value. The distance to move along pk can be found by approximately solving the following one dimensional minimization problem to find a step length α:

$$\min_{\alpha>0} f(x_k + \alpha p_k)$$

- Trust region
  - information gathered about f is used to construct a model function mk whose behavior near the current point xk is similar to that of the actual objective function f .

Search Directions for Line Search Methods

- **Steepest descent direction**

  - steepest descent direction $-\nabla$ fk is the most obvious choice for search direction for a line search method. It is intuitive; among all the directions we could move from xk , it is the one along which f decreases most rapidly.

$$f(x_k + \alpha p) = f(x_k) + \alpha p^T \nabla f_k + \frac{1}{2}\alpha^2 p^T \nabla^2 f(x_k + tp) p$$

  - The rate of change in f along the direction p at xk is simply the coefficient of α

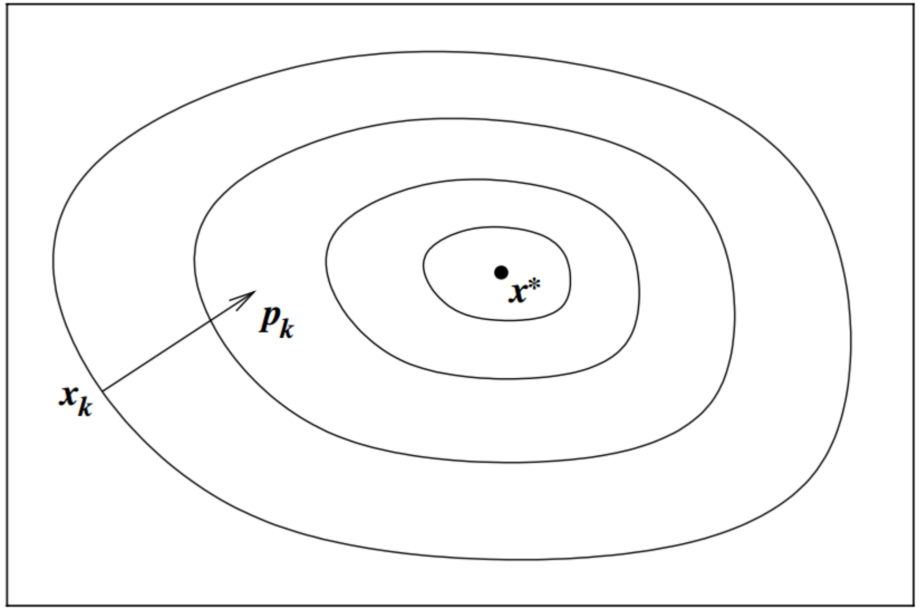
$$\min_p \; p^T \nabla f_k, \qquad \text{subject to } \|p\| = 1.$$

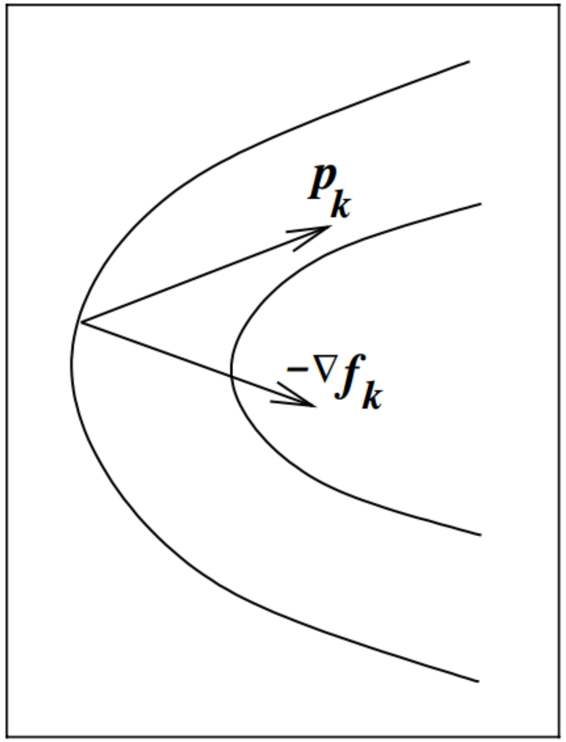
$$p^T \nabla f_k = \|p\| \, \|\nabla f_k\| \cos\theta = \|\nabla f_k\| \cos\theta$$

  - Cosine 이 -1이 될 때 제일 minimize가 된다.
  - Gradient만 계산해도 된다는 장점이 있지만, 복잡한 문제에서는 굉장히 느리다.

Search Directions for Line Search Methods

- **Steepest descent direction**
  - 단점을 때문에 steepest descent가 아니더라도 descent direction을 사용한다.





Search Directions for Line Search Methods

- **Descent direction**
  - In general, any descent direction—one that makes an angle of strictly less than π/2 radians with −∇ fk—is guaranteed to produce a decrease in f

$$f(x_k + \epsilon p_k) = f(x_k) + \epsilon p_k^T \nabla f_k + O(\epsilon^2).$$

  - Downhill direction이면

$$p_k^T \nabla f_k = \|p_k\| \, \|\nabla f_k\| \cos\theta_k < 0.$$

- **Newton direction**

$$f(x_k + p) \approx f_k + p^T \nabla f_k + \tfrac{1}{2} p^T \nabla^2 f_k p \stackrel{\text{def}}{=} m_k(p). \quad \nabla^2 f_k \text{ is positive definite,}$$

  - we obtain the Newton direction by finding the vector p that minimizes mk (p).

Search Directions for Line Search Methods

- **Newton direction**

$$f(x_k + p) \approx f_k + p^T \nabla f_k + \tfrac{1}{2} p^T \nabla^2 f_k p \stackrel{\text{def}}{=} m_k(p).\ \nabla^2 f_k \text{ is positive definite,}$$

- we obtain the Newton m direction by finding the vector p that minimizes k (p).
- 이 식을 미분해서 =0 으로 계산을 하면 아래와 같은 식이 나온다.

$$p_k^{\mathrm{N}} = -\left(\nabla^2 f_k\right)^{-1} \nabla f_k. \qquad (2.15)$$

- 이럴 경우 오차는 3차 항이기 때문에 p의 크기가 작은 경우 이러한 approximation은 꽤 정확하다고 할 수 있다.

$$\nabla f_k^T p_k^{\mathrm{N}} = -p_k^{\mathrm{N}\,T} \nabla^2 f_k p_k^{\mathrm{N}} \le -\sigma_k \| p_k^{\mathrm{N}} \|^2$$

- Gradient 가 0이 아닌 이상 descent directio이 된다.

- 그러나 이 방법을 사용할 때에는 Hessian $\nabla^2 f(x)$을 사용해야 하기 때문에 계산기 복잡할 수 있고, error가 일어날 수 있다.
- 그래서 Hessian을 새롭게 추정해서 대체하는 방법을 사용한다.

Search Directions for Line Search Methods

- **Newton direction**

$$f(x_k + p) \approx f_k + p^T \nabla f_k + \tfrac{1}{2} p^T \nabla^2 f_k p \stackrel{\text{def}}{=} m_k(p).\ \nabla^2 f_k \text{ is positive definite,}$$

- we obtain the Newton m direction by finding the vector p that minimizes k (p).
- 이 식을 미분해서 =0 으로 계산을 하면 아래와 같은 식이 나온다.

$$p_k^{\mathrm{N}} = -\left(\nabla^2 f_k\right)^{-1} \nabla f_k. \tag{2.15}$$

- 이럴 경우 오차는 3차 항이기 때문에 p의 크기가 작은 경우 이러한 approximation은 꽤 정확하다고 할 수 있다.

$$\nabla f_k^T p_k^{\mathrm{N}} = -p_k^{\mathrm{N}T} \nabla^2 f_k p_k^{\mathrm{N}} \le -\sigma_k \| p_k^{\mathrm{N}} \|^2$$

- Gradient 가 0이 아닌 이상 descent direction이 된다.

- 그러나 이 방법을 사용할 때에는 Hessian $\nabla^2 f(x)$을 사용해야 하기 때문에 계산기 복잡할 수 있고, error가 일어날 수 있다.
- 그래서 Hessian을 새롭게 추정해서 대체하는 방법을 사용한다.

Search Directions for Line Search Methods

Quasi Newton

- Hessian대신에 approximation B_k를 사용한다. which is updated after each step to take account of the additional knowledge gained during the step.
- The updates make use of the fact that changes in the gradient g provide information about the second derivative of f along the search direction.
- 기존의 Taylor's theorem에서 $\nabla^2 f(x)p$ 를 더했다가 뺀다

$$\nabla f(x+p) = \nabla f(x) + \nabla^2 f(x)p + \int_0^1 \left[\nabla^2 f(x+tp) - \nabla^2 f(x)\right] p\, dt.$$

$$\nabla f(x+p) = \nabla f(x) + \int_0^1 \nabla^2 f(x+tp)p\, dt,$$

- Because ∇ f (·) is continuous, the size of the final integral term is o(| p |).

$$x = x_k \qquad p = x_{k+1} - x_k$$

- 위처럼 설정을 하면 위의 식은 아래처럼 된다.

$$\nabla f_{k+1} = \nabla f_k + \nabla^2 f_k (x_{k+1} - x_k) + o(\|x_{k+1} - x_k\|).$$

Search Directions for Line Search Methods

- **Quasi Newton**
  - 이를 정리하면 $\nabla^2 f_k(x_{k+1} - x_k) \approx \nabla f_{k+1} - \nabla f_k.$
  - $s_k = x_{k+1} - x_k, \qquad y_k = \nabla f_{k+1} - \nabla f_k.$
  - $B_{k+1}s_k = y_k$ 라는 식이 나온다.

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k}, \qquad B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}.$$

  - Hessian 을 대체해서 사용 $p_k = -B_k^{-1} \nabla f_k$

$$H_k \stackrel{\text{def}}{=} B_k^{-1}$$

$$H_{k+1} = \left(I - \rho_k s_k y_k^T\right) H_k \left(I - \rho_k y_k s_k^T\right) + \rho_k s_k s_k^T, \qquad \rho_k = \frac{1}{y_k^T s_k}$$

Line Search

Step length

- Each iteration of a line search method computes a search direction pk and then decides how far to move along that direction.
- In computing the step length αk , we face a tradeoff. We would like to choose αk to give a substantial reduction of f , but at the same time we do not want to spend too much time making the choice.

$$\phi(\alpha) = f(x_k + \alpha p_k), \quad \alpha > 0,$$

- a의 candidate를 정한 다음 certain condition을 만족하는 a를 정한다

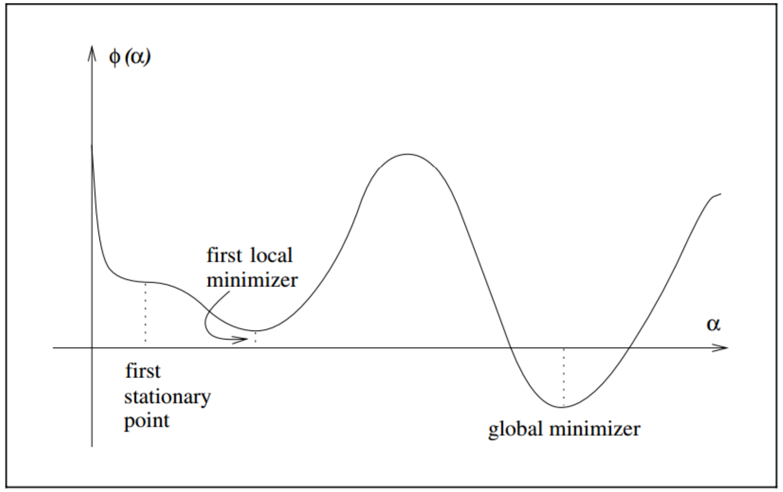
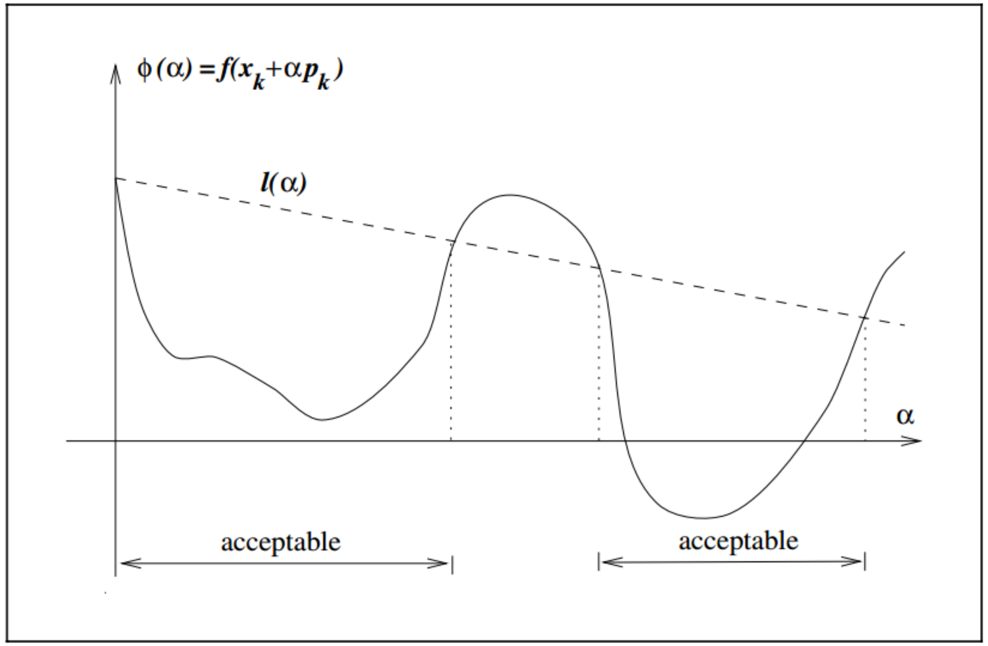


Figure 3.1 The ideal step length is the global minimizer.

Line Search

- **Wolfe conditions**
  - A popular inexact line search condition stipulates that αk should first of all give **sufficient decrease** in the objective function f , as measured by the following inequality:

$$f(x_k + \alpha p_k) \le f(x_k) + c_1 \alpha \nabla f_k^T p_k, \qquad c_1 \in (0, 1).$$



Line Search

- **Wolfe conditions**
  - Curvature condition $\nabla f(x_k + \alpha_k p_k)^T p_k \geq c_2 \nabla f_k^T p_k, \quad c_2 \in (c_1, 1),$
  - Note that the left-handside is simply the derivative φ (αk), so the curvature condition ensures that the slope of φ at αk is greater than c2 times the initial slope φ (0).