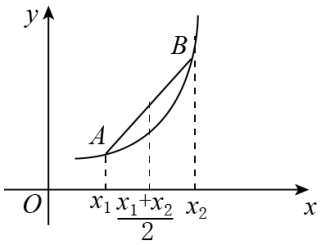
凸优化讲义

0. 数学基础

0.1 拉格朗日中值定理

$$f(x_2)-f(x_1)=f'(\xi)(x_2-x_1) \quad (x_1<\xi<x_2)$$



应用：使用微分中值定理证明函数单调性。

0.2 泰勒展开

$$f(x)=f(x_0)+f'(x_0)(x-x_0)+\frac{f''(x_0)}{2!}(x-x_0)^2+\ldots+\frac{f^{(n)}(x_0)}{n!}(x-x_0)^n+R_n(x)$$

其中，$f^{(n)}(x)$ 表示 f（x）的 n 阶导数，等号后的多项式称为函数 f（x）在 x_0 处的泰勒展开式，剩余的 R_n（x）是泰勒公式的余项，是$(x-x_0)$的高阶无穷小。

应用：凸函数可以用泰勒展开进行逼近。

0.3 向量微积分

Jacobian 矩阵：

$$\mathbf{J}=\begin{bmatrix}\frac{\partial \mathbf{f}}{\partial x_1} & \cdots & \frac{\partial \mathbf{f}}{\partial x_n}\end{bmatrix}=\begin{bmatrix}\frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n}\\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n}\end{bmatrix}$$

$\mathbf{y}=\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x}+\mathbf{B}\mathbf{x},\mathbf{A}$为对称阵，求$\frac{d\mathbf{y}}{d\mathbf{x}}$？ $\frac{d\mathbf{y}}{d\mathbf{x}}=2\mathbf{A}\mathbf{x}+\mathbf{B}^T$

0.4 半正定矩阵

对于任何非零向量 $\boldsymbol{x}$，$\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x}\geq 0$，则 $\boldsymbol{A}$ 为半正定，

应用：多元函数的二阶偏导矩阵是 A，如何证明它是凸的：矩阵 A 是半正定的

0.5 梯度

数学符号为▽，多元函数的梯度是一阶偏导数构成的一个向量。

$\nabla f(X^{(0)})=\left[\frac{\partial f}{\partial x_1},\frac{\partial f}{\partial x_2},\ldots,\frac{\partial f}{\partial x_n}\right]\Big|_{X^{(0)}}^{T}$，它是 $f(X)$ 在 $X^{(0)}$ 点处的梯度。

0.6 Hessian 矩阵

多元函数的二阶偏导数构成的矩阵为 Hessian 矩阵。

$$G(X^{(0)})=\left[\begin{array}{cccc}\frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2}\end{array}\right]_{(X^{(0)})}$$ 为函数 $f(X)$ 在 $X^{(0)}$ 点处的 Hessian 矩阵。Hessian 矩阵是由目标函数 f 在点 x 处的二阶偏导数组成的 $n\times n$ 阶矩阵。

如果函数 f 的二阶偏导可交换，则 Hessian 矩阵是对称矩阵。

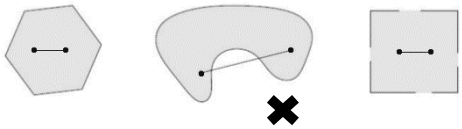
1. 凸集

1.1 凸集的定义

定义：集合 C 内任意两点间的线段均在集合 C 内，则称集合 C 为凸集。

$\forall x_1, x_2 \in C, \theta \in [0,1]$, 则 $\theta x_1 + (1-\theta)x_2 \in C$

例：



n维实数空间：R^n, 仿射子空间:$\{x \mid Ax = b\}$

所有半正定矩阵构成的集合是凸集

1.2 凸集的性质

凸集的∩也是凸集

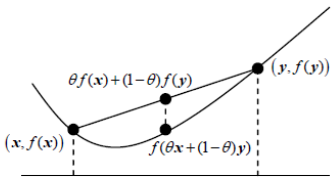
2.凸函数

2.1 凸函数的定义

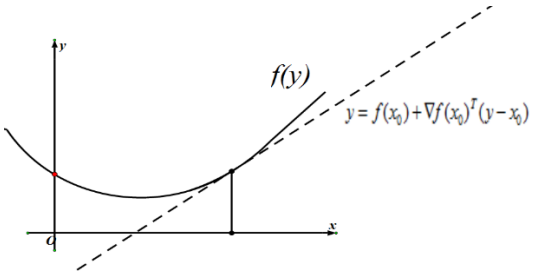
定义：若函数 f 的定义域 dom（f），f 为凸集，且满足

$\forall x, y \in dom(f), 0 \le \theta \le 1$，有

$$f(\theta x + (1-\theta)y) \le \theta f(x) + (1-\theta) f(y)$$



一阶条件：若函数 f 的定义域 dom（f）为开集，且函数 f 一阶可微，则函数 f 为凸函数当前仅当 dom（f）为凸集，且 $\forall x, y \in dom(f)$, $f(y) \ge f(x) + \nabla f(x)^T (y-x)$。



二阶条件：若函数f的定义域 dom（f）为凸集，且函数f二阶可微，则函数f为凸函数当前仅当 dom（f）为凸集，且$\nabla^2 f(x)\geq 0$。

2.2 凸函数举例

（1） $f(x)=e^{ax},x\in R$

（2） $f(x)=-\log x$，-log（f（x））常见于损失函数的设计

（3） $f(x)=\mathbf{q}^{\mathbf{T}}\mathbf{x}+r(q\in R^n,r\in R)$

（4） $f(x)=\frac{1}{2}x^TPx+q^Tx+r([x\in R^n，\ P\in R^{n\times n},q\in R^n,r\in R)$

（1）证明： $f(x)=e^{ax},x\in R$ ， $f'(x)=e^{ax}a$ ， $f''(x)=e^{ax}a^2\geq 0$

故指数函数是凸函数。

（2）证明： $f(x)=-\log x$， $f(x)'=-\frac{1}{x}$ ， $f(x)''=\frac{1}{x^2}>0$

故函数为凸函数。

（3）证明: $f(x)=q^Tx+r(q\in R^n,r\in R)$ $\nabla^2 f(x)=0$

（4）证明: $f(x)=\frac{1}{2}x^TPx+q^Tx+r([x\in R^n，\ P\in R^{n\times n}$且对称$,q\in R^n,r\in R)$

$\nabla^2 f(x)=P,P$为半正定矩阵时，函数为凸函数。

3. 凸优化问题

3.1 凸优化问题的标准形式

$$\min\ f(x)$$

$$s.t.\ h_i(x)\leq 0\quad i=1,...,m$$

$$\ell_j(x)=0\quad j=1,...,r$$

其中$x\in R^n$：优化变量； f：目标函数/损失函数； $h_i(x)\leq 0$：不等式约

束；h_i：不等式约束函数；$\ell_j(x)=0$ ：等数约束；ℓ_j：等式约束函数；如果没有约束（即 $m=p=0$）我们称问题为无约束问题；定义域：

$D=\bigcap_{i=1}^{m} dom(h)_i \cap \bigcap_{j=1}^{p} dom(\ell_j)$；可行解空间：$X=\left\{x\in D \middle| \begin{matrix} h_i(x)\le 0 & i=1,\cdots,m \\ \ell_j(x)=0 & j=1,\cdots,r \end{matrix}\right\}$

（目标函数是凸的，不等式约束函数是凸的，等式约束函数必须是仿射的，可行解空间是凸集）

3.2 对偶问题

Given a minimization problem

$$\min_{x\in\mathbb{R}^n} f(x)$$
$$\text{subject to } h_i(x)\le 0,\ \ i=1,\cdots m$$
$$\ell_j(x)=0,\ j=1,\cdots r$$

we defined the Lagrangian：

$$L(x,u,v)=f(x)+\sum_{i=1}^{m}u_i h_i(x)+\sum_{j=1}^{r}v_j\ell_j(x)$$

and Lagrange dual function：

$$g(u,v)=\inf_{x\in D} L(x,u,v)$$

其中 u,v 为**拉格朗日乘子**，也叫对偶变量，顾名思义，u,v 是对偶问题中的变量。若拉格朗日函数没有下界，则 $g(u,v)=-\infty$。由于对偶函数是一族关于 (u,v) 的仿射函数的逐点下确界，即使原问题不是凸的，对偶函数也是凹函数。

The subsequent **dual problem** is：

$$\max_{u\in\mathbb{R}^m,v\in\mathbb{R}^r} g(u,v)$$
$$\text{subject to } u\ge 0$$

Important properties：

- Dual problem is always convex,i.e.,g is always concave(even if primal problem is not convex)
- The primal and dual optiomal values, f^* and g^*,always satisfy weak duality:$f^*\ge g^*$
- Slater's condition:for convex primal,if there is an x such that $h_1(x)<0,\cdots h_m(x)<0$ and $\ell_r(x)=0,\cdots\ell_r(x)=0$ then strong duality holds:$f^*=g^*$.(Can be further refined to strict inequalities over nonaffine $h_i, i=1,\cdots m$)

3.3 KKT 条件

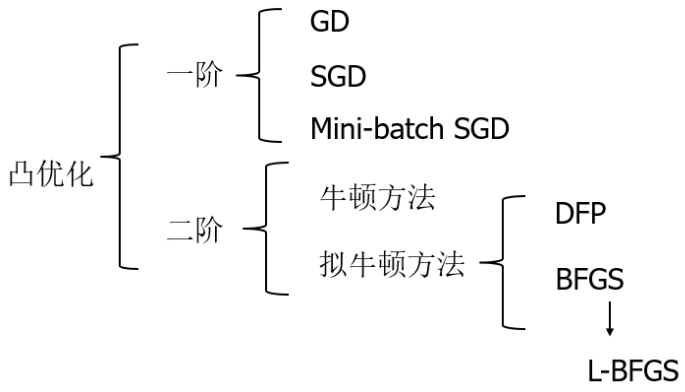
Given general problem

$$\min_{x\in\mathbb{R}^n} f(\mathrm{x})$$
$$\text{subject to } h_i(x)\le 0, i=1,\cdots m$$
$$\ell_j(x)=0,\ j=1,\cdots r$$

The **Karush-Kucher conditions** or KKT conditions are:

- $u_i\cdot h_i(x)=0$ for all i
- $h_i(x)\le 0, \ell_j(x)=0$ for all i, j (conplementary slackness)
- $h_i(x)\le 0, \ell_j(x)=0$ for all i, j (primal feasibility)
- $u_i\ge 0$ for all i (dual feasibility)

4.凸优化算法



$\nabla f(X^{(0)})=\left[\frac{\partial f}{\partial x_1},\frac{\partial f}{\partial x_2},...,\frac{\partial f}{\partial x_n}\right]\Big|^T_{X^{(0)}}$，它是 $f(X)$ 在 $X^{(0)}$ 点处的梯度。

$$G(X^{(0)})=\begin{bmatrix}\frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1\partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1\partial x_n}\\ \frac{\partial^2 f}{\partial x_2\partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2\partial x_n}\\ \vdots & \vdots & \ddots & \vdots\\ \frac{\partial^2 f}{\partial x_n\partial x_1} & \frac{\partial^2 f}{\partial x_n\partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2}\end{bmatrix}_{(X^{(0)})}$$

为函数 $f(X)$ 在 $X^{(0)}$ 点处的 Hessian 矩阵。

4.1 一阶方法

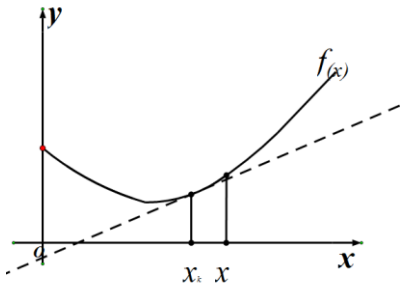
4.1.1 GD

梯度下降形式：$w_{k+1}=w_k-\lambda\nabla_w f(w)$

一阶泰勒展开：

$$f(x)=f(x_k)+\nabla f(x_k)^T\cdot(x-x_k)+o(\|x-x_k\|^2)$$
$$\approx f(x)=f(x_k)+\nabla f(x_k)^T\cdot(x-x_k)$$

可以写为：$\varphi(x)=f(x_k)+\nabla f(x_k)^T(x-x_k)$



问题：其中 $f(x_k)$ 为常数， $f'(x_k)$ 为一阶导数，$(x-x_k)$ 如何取值 $\varphi(x)$ 最小？

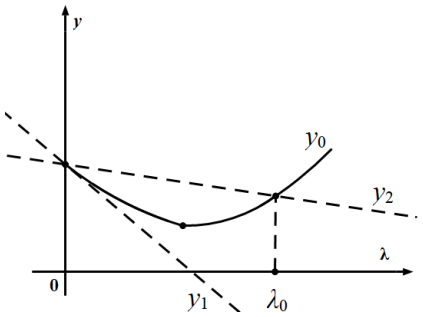
两个向量相乘：$\vec{a}\cdot\vec{b}=|\mathbf{a}||\mathbf{b}|\cos\theta$，因此 $(x-x_k)$ 为 $\nabla f(x_k)$ 反方向的时候，$\varphi(x)$ 最小。此时 $x-x_k=-\nabla f(x_k)$，将 x 设为 x_{k+1} 得到：$x_{k+1}=x_k-\nabla f(x_k)$，$\Delta x=-\nabla f(x_k)$。

在实际当中我们不会每次移动梯度那么大，因此我们选择一个 λ，$0<\lambda<1$，得到：$x_{k+1}=x_k-\lambda\nabla f(x_k)$。通过步长 λ 的调整可以尽快收敛，那么步长 λ 如何选择？下面我们以一阶为例：

问题：

$\min f(x_k+\lambda\cdot\Delta x)$，$\lambda>0$

取 $\alpha\in(0,0.5)$，在书中 P445 有具体说明 α 为何 0 到 0.5 之间，在图中画出 $y_0=f(x_k+\lambda\cdot\Delta x)$ 函数，确定 $y_1=f(x_k)+\lambda\nabla f(x_k)^T\Delta x$（切线），$y_2=f(x_k)+\alpha\lambda\nabla f(x_k)^T\Delta x$ 两条直线，其中 y_2 与函数的交点的横坐标设为 λ_0，λ 的搜索条件是 y_0 位于两条线之间。



我们采用回溯搜索的方法，取 $\lambda=1$ 开始搜索，$\lambda:=\beta*\lambda$，$0<\beta<1$，每次缩小都计算函数 $f(x_k+\lambda\cdot\Delta x)$ 的值直到搜索到最佳步长。

4.1.2 SGD

由于 GD 算法速度慢，并且需要超大量的数据内存，因此随机选择单个样本进行梯度计算，迭代方式为：$w_{t+1} = w_t - \eta_t \nabla_w L(w, x_i, y_i)$。

存在的问题：方差大，损失函数震荡严重。

4.1.3 Mini-batch SGD

综合 GD 与 SGD 的优缺点，我们采用 Mini-batch SGD 的方法，对随机 mini-batch 计算梯度，进行参数更新。更新方式为：

$$w_{t+1} = w_t - \frac{1}{N}\eta_t \sum_{i=1}^{N} \nabla_w L(w_t, x_i, y_i)$$

Mini-batch SGD:

for *i* = 1 : *Num_Iterations*

1. 随机采样一个 Mini-batch 的样本
2. 前向操作：计算损失
3. 后向操作：通过 BP 算法计算梯度
4. 网络更新：利用步骤 3）计算的梯度更新网络参数

end

Weight Decay：

Weight Decay（权值衰减）的使用既不是为了提高你所说的收敛精确度也不是为了提高收敛速度，而是为了避免过拟合，放在正则项（Regularization）前面的一个系数：

$$\tilde{L}(w) = L(w) + \frac{\lambda}{2}\|w\|_2^2$$

在实际应用中，二范数较常用，一般λ设置的较小，例如 0.0005。

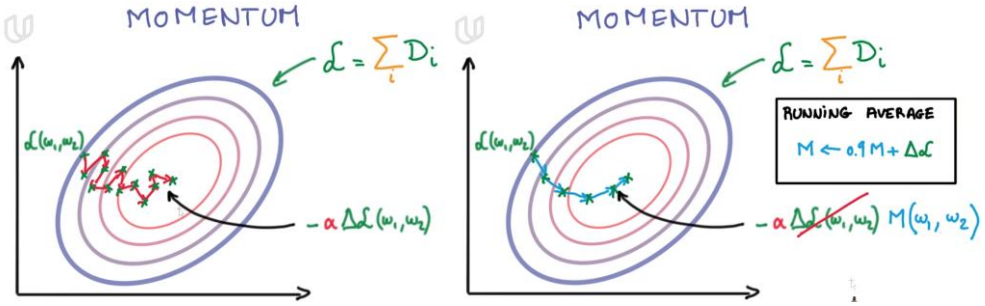
Momentum：

计算梯度时考虑历史梯度信息，使随机梯度下降更容易跳出局部最优 ，加速收敛：

$$v_{t+1} = \mu v_t + \eta_t \nabla L(w_t)$$
$$w_{t+1} = w_t - v_{t+1}$$

其中 v 为梯度的移动平均（moving average）形式，在实际应用中我们一般

取 $\mu = 0.9$。



图片来自：

https://classroom.udacity.com/courses/ud730/lessons/6370362152/concepts/63798118400923#

4.2 二阶方法

4.2.1 牛顿法

泰勒展开： $\varphi(x)=f(x_k)+f'(x_k)(x-x_k)+\frac{1}{2}f''(x_k)(x-x_k)^2+o(x-x_k)^3$

$\varphi(x)=f(x_k)+f'(x_k)(x-x_k)+\frac{1}{2}f''(x_k)(x-x_k)^2$（略去高阶项）

求导：$\varphi'(x)=f'(x_k)+f''(x_k)(x-x_k)$

在达到极值时 $\varphi'(x)=0$ 则 $x=x_k-\frac{f'(x_k)}{f''(x_k)}$，得到迭代方式：$x_{k+1}=x_k-\frac{f'(x_k)}{f''(x_k)}$，

该方法称之为**牛顿法**。

拓展到向量形式：

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

两边取梯度，在极小值条件下：

$$g_k+H_k(x-x_k)=0$$

得到：$x=x_k-H_k^{-1}g_k$

写成迭代形式：$x_{k+1}=x_k-H_k^{-1}g_k$

其中： $\nabla f(x_k)$ 记做 $g(x_k)$ 简写为 g_k；$\nabla^2 f(x)$ 记做 $H(x_k)$ 简写为 H_k

牛顿法过程：

1. 初始 x_0，阈值 ε，令 $k:=0$
2. 计算 g_k 和 H_k
3. 若 $\|g_k\|<\varepsilon$，停止，否则 $d_k=-H_k^{-1}g_k$
4. $x_{k+1}=x_k+\lambda d_k$（确定最优步长）
5. $k:=k+1$，到 2

缺陷：需要算 Hessian 矩阵计算量大，因此引入拟牛顿法，核心思想是泰勒展开进行近似，对近似值进行分析。

4.2.2 拟牛顿法

泰勒展开：$f(x)\approx f(x_{k+1})+\nabla f(x_{k+1})^T(x-x_{k+1})+\frac{1}{2}(x-x_{k+1})^T\nabla^2 f(x_{k+1})(x-x_{k+1})$

两边分别求梯度，得到：$\nabla f(x)\approx\nabla f(x_{k+1})+H_{k+1}(x-x_{k+1})$

令：$x=x_k$，则 $g_{k+1}-g_k\approx H_{k+1}(x_{k+1}-x_k)$

引入符号：$s_k=x_{k+1}-x_k, y_k=g_{k+1}-g_k$

$$\begin{cases} y_k\approx H_{k+1}\cdot s_k \leftarrow \text{BFGS方法} \\ s_k\approx H_{k+1}^{-1}\cdot y_k \leftarrow \text{DFP方法} \end{cases}$$

DFP 方法:

如果 Hessian 矩阵偏导可交换那么它是实对阵矩阵，我们可以以此迭代求解，采用 D_{k+1} 去逼近 H_{k+1}^{-1} 进行近似，即 $H_{k+1}^{-1}\approx D_{k+1}$。

我们将 D_{k+1} 定义为：$D_{k+1}=D_k+\Delta D_k, k=0,1,2,\ldots$，$\Delta D_k=\alpha\mu\mu^T+\beta vv^T$，这种形式保证了 D_{k+1} 始终是对称的。

我们将 D_{k+1}，带入 $s_k\approx H_{k+1}^{-1}\cdot y_k$：

$$s_k\approx H_{k+1}^{-1}y_k\Rightarrow s_k=D_{k+1}y_k=D_ky_k+\alpha\mu\mu^Ty_k+\beta vv^Ty_k$$

$$\begin{aligned} s_k&=D_ky_k+\mu(\alpha\mu^Ty_k)+v(\beta v^Ty_k) \\ &=D_ky_k+(\alpha\mu^Ty_k)\mu+(\beta v^Ty_k)v \end{aligned}$$

假设：$\alpha\mu^T y_k = 1$，$\beta v^T y_k = -1$

解得：$\alpha = \frac{1}{\mu^T y_k}$，$\beta = \frac{-1}{v^T y_k}$

得到：$\mu - v = s_k - D_y y_k$

我们可以找到一个解：$\begin{cases} \mu = S_k \\ v = D_k y_k \end{cases}$

由此可以解出α，β，那么$D_{k+1} = D_k + \Delta D_k$可解，避免了求 Hessian 矩阵。

$$\alpha = \frac{1}{S_K^T y_k}, \beta = -\frac{1}{(D_k y_k)^T y_k} = -\frac{1}{y_k^T D_k y_k},$$

$$\Delta D_k = \frac{S_k S_k^T}{S_k^T y_k} - \frac{D_k y_k D_k}{y_k^T D_k y_k}$$

5. 参考资料

材料一：支持向量机

材料二：对偶问题与 KKT 条件

材料三：凸函数判定的一阶和二阶条件

材料四：凸优化方法：http://blog.csdn.net/itplus/article/details/21896453

作业:

1.Sigmod 函数 $S(x)=\frac{1}{1+e^{-x}}$ 是不是凸的？

2. 为什么说深度网络是非凸的？

3. 为什么不用二阶方法来优化深度网络？

4. 什么是鞍点？对于深度网络优化来如何跳出鞍点？