**二阶优化算法**

考虑无约束优化问题

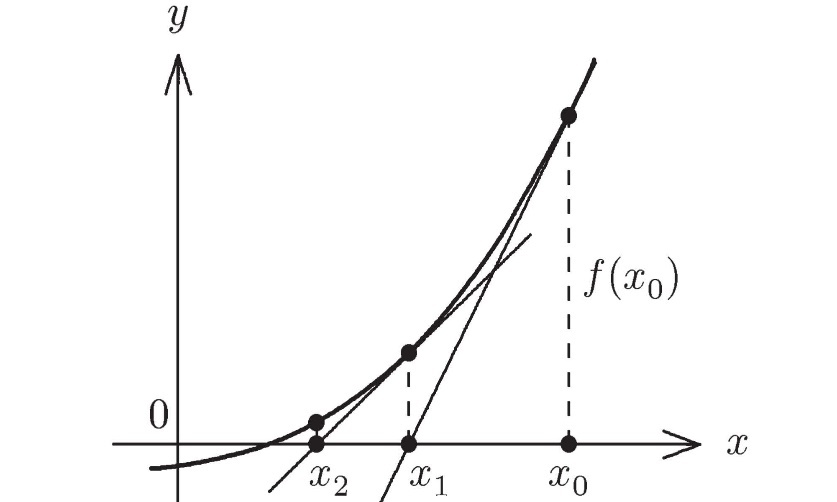
假设具有二阶连续导数

是的梯度向量在点的值

是的黑塞矩阵在点的值

**牛顿法(Newton’s method)**

**一元形式(切线法)**



求的根

选取作为的初始近似值，过点做曲线的切线

切线与轴交点

重复以上过程，得到

称为的第次迭代近似值

将一元的切线法拓展到多元，并求导函数等于0的根即得到用于优化的牛顿法

**算法步骤**

输入：目标函数，梯度，黑塞矩阵，精度要求

输出：最小点的近似值

(1)置，选择初始点  
(2)计算，若，则停止计算，得到近似解

(3)计算

(4)由求出

(5)置

(6)令，转(2)

**核心思想**

若第次迭代值为，则可将在附近进行二阶泰勒展开：

有极值的必要条件是在极值点处一阶导数为，即梯度向量为，特别是当为正定矩阵时，的极值为极小值

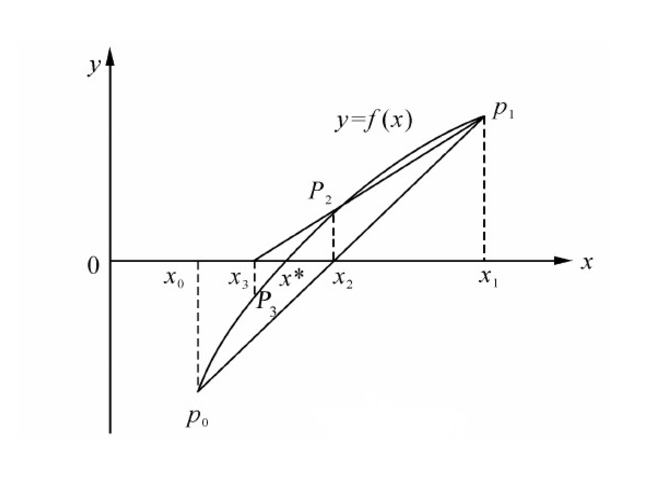
牛顿法每次迭代从开始，利用极小值必要条件求目标函数的极小点，作为第次迭代值

二阶泰勒展开式求导得

假设，则

故

**拟牛顿法(Quasi-Newton method)**

**一元形式(割线法)**

求的根

第次迭代时，已知上两点和

过这两个点的割线为

将代入割线方程解得

对上式进行反复迭代即可得到根的近似值

割线法的优势是无需精确计算导数

将一元的割线法拓展到多元，并求导函数等于0的根即得到用于优化的拟牛顿法

**核心思想**

在牛顿法的迭代中，要计算黑塞矩阵的逆矩阵，这一计算比较复杂，拟牛顿法用一个阶矩阵来近似代替

牛顿法迭代中黑塞矩阵满足条件

记，

故满足的条件也可写成或

牛顿法迭代公式中没有步长因子，且可能非正定，因此有时会出现函数值上升，即的情况，甚至最终发散。如果正定(等价于正定)，那么取小步长因子可以保证牛顿法搜索方向是下降方向

证明：

令

在附近的一阶泰勒展开式可以近似写成

因正定，故当充分小时，总有，也就是说是下降方向

拟牛顿法将作为的近似，要求满足下面的条件

第一，每次迭代矩阵是正定的

第二，

说明：

设为目标函数的一阶导数，在点处

由切线法，，故

由割线法，，故

拟牛顿法的关键在于每一步迭代中矩阵的更新，此步有多种实现方法

**BFGS(Broyden-Fletecher-Goldforb-Shanno)**

BFGS算法是目前最流行的拟牛顿算法

**算法步骤**

输入：目标函数，梯度，精度要求

输出：最小点的近似值

(1)置，选定初始点，初始化正定对称矩阵

(2)计算，若，则停止计算，得到近似解，否则转(3)

(3)求出

(4)一维搜索：求出使得也许要先给一个范围

(5)置

(6) 计算, 若，则停止计算，得到近似解计算

(7)令

**核心思想**

BFGS可以考虑用逼近黑塞矩阵，也可以考虑用逼近黑塞矩阵的逆矩阵

更新公式为

说明：

总满足

若初始矩阵是正定的，则迭代过程中每个矩阵都是正定的

若记

则更新公式可变为

称为BFGS算法关于的迭代公式

**LBFGS(Limited-memory BFGS)**

**大规模问题**

设对于每个样本损失函数为，其中是预测值，与样本的特征相关，是标签

优化模型可写为

第一类大规模问题：很大，每次迭代运算量大且冗余较多，按整个数据集上的Epoch来看收敛较慢，可通过取小批量解决

第二类大规模问题：很大，即每个样本的特征数较多，此时黑塞矩阵较大，需要占用大量的存储资源，很多算法不再奏效，LBFGS就是为了解决此问题，减少迭代过程中的内存开销

**算法步骤**

输入：目标函数，迭代初始值，存储长度

输出：最小点的近似值

**while** not accurate enough **do**

Obtain the search direction by two-loop recursion

with satisfying the Wolfe condition

and

and

Add into storage

**if** **then** *#丢弃后的序列是为下一次迭代更新准备的*

discard vector pair and scalar from storage

**else**

**end if**

**end while**

return

神经风格迁移中，优化目标是有固定损失函数的样本，而非通过样本或小批量更新神经网络参数，因此非常适合LBFGS

**核心思想**

**1、算法纲要**

在BFGS中，，需要组数据

在LBFGS中，只保存最近组数据，舍弃更靠后的，

即，并找一个已知矩阵代替

在BFGS更新公式中

令，，则

依此从已知的开始迭代，一直迭代到

常令

**2、计算更新值**

第次迭代中，要选取的搜索方向是

实际上，无需计算，即可以通过two-loop recursion算法求得

下面通过该算法求

输入：目标函数,，，，

输出：

**for** **do**

**end for**

**for** **do**

**end for**

return

可以用一个例子说明：知，，求

先用迭代公式求，再用two-loop recursion求，验证出二者相等

由此，在LBFGS算法中，不再保存完整的，而是存储最新步的向量序列和(若向量序列全部存储，则计算结果与BFGS相同)，需要矩阵时，使用向量序列和计算就可以得到

**3、的选取**

此文中不讨论该问题