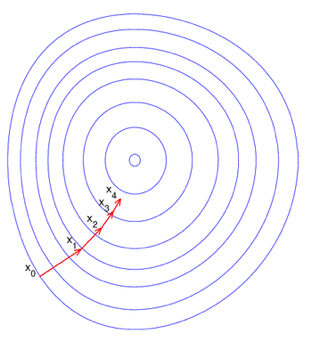
**Optimization**

**1. 说说梯度下降法。**

梯度下降法是最早最简单，也是最为常用的最优化方法。梯度下降法实现简单，当目标函数是凸函数时，梯度下降法的解是全局解。一般情况下，其解不保证是全局最优解，梯度下降法的速度也未必是最快的。梯度下降法的优化思想是用当前位置负梯度方向作为搜索方向，因为该方向为当前位置的最快下降方向，所以也被称为是”最速下降法“。最速下降法越接近目标值，步长越小，前进越慢。梯度下降法的搜索迭代示意图如下图所示：



梯度下降法的缺点：靠近极小值时收敛速度减慢；可能会“之字形”下降。

**2. 梯度下降法找到的一定是下降最快的方向么？**

梯度下降法并不是下降最快的方向，它只是目标函数在当前的点的切平面上下降最快的方向。

**3. 梯度下降方法的变体有哪些？**

|  |  |  |
| --- | --- | --- |
| 梯度下降方法 | 描述 | 特点 |
| 批量梯度下降算法BGD | 在更新参数时使用所有的样本计算梯度来进行更新 | 优点：全局最优解；易于并行实现；  缺点：当样本数目很多时，训练过程会很慢。 |
| 随机梯度下降算法SGD | 在更新参数时随机使用一个样本计算梯度来进行更新 | 优点：训练速度快；  缺点：准确度下降，并不是全局最优；不易于并行实现。 |
| 小批量梯度下降法MBGD | 在更新参数时随机选取一小批样本计算梯度来进行更新 | BGD和SGD的折衷，MBGD的训练过程比较快，而且保证最终参数训练的准确率。 |

**4. 介绍下牛顿法。**

牛顿法是一种在实数域和复数域熵近似求解方程的方法。方法使用函数的泰勒级数的前面几项来寻找方程的根。牛顿法最大的特点在于它的收敛速度很快。

为了求解 的根，把的泰勒展开到二阶形式：

上述式子成立，当且仅当无限趋近于0，此时上式等价于，求得，得出迭代公式：

上述对应的是二维的情况，高维的牛顿公式是

 其中为hessian矩阵，定义为：

牛顿法的优缺点总结：

优点：二阶收敛，收敛速度快

缺点：对目标函数有较严格的要求，函数必须具有连续的一二阶偏导数，海森矩阵必须正定；牛顿法是一种迭代算法，每一步都需要求解目标函数的Hessian矩阵的逆矩阵，计算比较复杂

**5. 牛顿法和梯度下降法有什么不同？**

梯度下降法用目标函数的一阶偏导、以负梯度方向作为搜索方向，只考虑目标函数在迭代点的局部性质；牛顿法同时考虑了目标函数的一、二阶偏导数，考虑了梯度变化趋势，因而能更合适的确定搜索方向加快收敛。从几何上说，牛顿法就是用一个二次曲面去拟合你当前所处位置的局部曲面，而梯度下降法是用一个平面去拟合当前的局部曲面，通常情况下，二次曲面的拟合会比平面更好，所以牛顿法选择的下降路径会更符合真实的最优下降路径。

**6. 什么是拟牛顿法？**

设经过k+1次迭代后得到，此时将目标函数在附近作泰勒展开，取二阶近似，得到

在上式两边同时作用一个梯度算子，可得

(

对上式中取并整理，可得

若引入记号，则上式可紧凑地写成

或者。

这就是所谓的拟牛顿条件，它对迭代过程中的海森矩阵做约束，因此，对作近似的，以及对作近似的可以将或者作为指导。

DFP算法是拟牛顿法的一种方法，其核心是，通过迭代的方法，对作近似，迭代格式为：

其中的通常取为单位矩阵。因此，关键是每一步的矫正矩阵如何构造。首先，采用待定法，构造出下式：

其中，和为待定系数，和为待定向量，从形式上看，这种待定公式至少保证了矩阵的对称性。然后，将上式代入，可得

再进行改写，得到

括号中的和是两个数，假设

带回原式，得到

取，，可得

其中，第二个等式用到了的对称性。最后，综合上面的推导，可以得出

**7. 请说说随机梯度下降法的问题和挑战。**

①问题

在每次更新时用1个样本，可以看到多了随机两个字，随机也就是说我们用样本中的一个例子来近似所有的样本，来调整θ，因而随机梯度下降是会带来一定的问题，因为计算得到的并不是准确的一个梯度，对于最优化问题，凸问题，虽然不是每次迭代得到的损失函数都向着全局最优方向， 但是大的整体的方向是向全局最优解的，最终的结果往往是在全局最优解附近。但是相比于批量梯度，这样的方法更快，更快收敛，虽然不是全局最优，但很多时候是我们可以接受的。SGD的问题在于由于频繁的更新和波动，最终将收敛到最小限度，并会因波动频繁存在超调量。

②挑战

很难选择出合适的学习率。太小的学习率会导致网络收敛过于缓慢，而学习率太大可能会影响收敛，并导致损失函数在最小值上波动，甚至出现梯度发散；

.此外，相同的学习率并不适用于所有的参数更新。如果训练集数据很稀疏，且特征频率非常不同，则不应该将其全部更新到相同的程度，但是对于很少出现的特征，应使用更大的更新率；

在神经网络中，最小化非凸误差函数的另一个关键挑战是避免陷于多个其他局部最小值中。实际上，问题并非源于局部极小值，而是来自鞍点，即一个维度向上倾斜且另一维度向下倾斜的点。这些鞍点通常被相同误差值的平面所包围，这使得SGD算法很难脱离出来，因为梯度在所有维度上接近于零。

**8. 有哪些梯度下降方法？**

①批量梯度下降法（BGD）：在更新参数时使用所有的样本来进行更新。

②随机梯度下降（SGD）：随机选取一个点做梯度下降，而不是遍历所有样本后进行参数迭代。因为梯度下降法的代价函数计算需要遍历所有样本，而且是每次迭代都要遍历，直至达到局部最优解，在样本量庞大时就显得收敛速度比较慢了，计算量非常庞大。

③小批量梯度下降（MBGD）：将随机采样一个批量样本进行计算梯度。

④动量（Momentum）：通过速度v，来积累了之间梯度指数级衰减的平均，并且继续延该方向移动，减少振荡过程。



⑤Nesterov梯度加速法：先根据之前的动量进行大步跳跃，然后计算梯度进行校正，从而实现参数更新。这种预更新方法能防止大幅振荡，不会错过最小值，并对参数更新更加敏感。

⑥Adagrad：能够在训练中自动的对learning rate进行调整，对于出现频率较低参数采用较大的α更新；相反，对于出现频率较高的参数采用较小的α更新。因此，Adagrad非常适合处理稀疏数据。

⑦RMSprop：Adagrad会累加之前所有的梯度平方，而RMSprop仅仅是计算对应的平均值，因此可缓解Adagrad算法学习率下降较快的问题。

⑧Adam(Adaptive Moment Estimation)：是另一种自适应学习率的方法。它利用梯度的一阶矩估计和二阶矩估计动态调整每个参数的学习率。Adam的优点主要在于经过偏置校正后，每一次迭代学习率都有个确定范围，使得参数比较平稳。

**9. 说说共轭梯度法。**

共轭梯度法是介于最速下降法与牛顿法之间的一个方法，它仅需利用一阶导数信息，但克服了最速下降法收敛慢的缺点，又避免了牛顿法需要存储和计算海森矩阵并求逆的缺点，共轭梯度法不仅是解决大型线性方程组最有用的方法之一，也是解大型非线性最优化最有效的算法之一。

在数值线性代数中，共轭梯度法是一种求解对称正定线性方程组的迭代方法。事实上，求解等价于求解，将其展开后可以得到，也就是等价于求解 。于是解方程问题就转化为了求解二次规划问题。共轭梯度法是介于梯度下降法与牛顿法之间的一个方法，是一个一阶方法。它克服了梯度下降法收敛慢的缺点，又避免了存储和计算牛顿法所需要的二阶导数信息。

共轭梯度法的思想就是找到个两两共轭的共轭方向，每次沿着一个方向优化得到该方向上的极小值，后面再沿其它方向求极小值的时候，不会影响前面已经得到的沿哪些方向上的极小值，所以理论上对个方向都求出极小值就得到了维问题的极小值。

算法推导过程如下：

目标函数的标准形式：

: 对于正定矩阵，如果非零向量，是的，那么。我们需要找到个相互的基向量，它们相互共轭且线性无关。因此空间中任意向量x都可以用这组基向量表示：。

因此我们的目标函数可以改写为：

因为是，所以，上式可以简化为

将求和符号提出来后，可以得到

原本我们的问题是寻找一个向量x，使得目标函数值最优。现在我们用来表示，也就是我们的任务变为优化了。把求和分开来写：

这样，我们分别求第一项的最小值，然后第二项，即可。

对上式进行求导，得，所以最优解，将上式代入可得

现在，我们只需要构建向量组。有一种称为Gram-Schmidt过程的算法可以用来找到这组向量。

**10. 优化方法有哪些？**

|  |  |
| --- | --- |
| 算法 | 描述 |
| 梯度下降法 | 梯度下降法的优化思想是用当前位置负梯度方向作为搜索方向，因为该方向为当前位置的最快下降方向，所以也被称为是”最速下降法“。最速下降法越接近目标值，步长越小，前进越慢。 |
| 牛顿法 | 牛顿法的基本思想是利用迭代点处的一阶导数(梯度)和二阶导数(Hessen矩阵)对目标函数进行二次函数近似，然后把二次模型的极小点作为新的迭代点，并不断重复这一过程，直至求得满足精度的近似极小值。牛顿法的速度相当快，而且能高度逼近最优值。 |
| 共轭梯度法 | 共轭梯度法是介于最速下降法与牛顿法之间的一个方法，它仅需利用一阶导数信息，但克服了最速下降法收敛慢的缺点，又避免了牛顿法需要存储和计算海森矩阵并求逆的缺点，共轭梯度法不仅是解决大型线性方程组最有用的方法之一，也是解大型非线性最优化最有效的算法之一。 在各种优化算法中，共轭梯度法是非常重要的一种。其优点是所需存储量小，具有步收敛性，稳定性高，而且不需要任何外来参数。 |
| 启发式优化方法 | 启发式方法指人在解决问题时所采取的一种根据经验规则进行发现的方法。其特点是在解决问题时,利用过去的经验,选择已经行之有效的方法，而不是系统地、以确定的步骤去寻求答案。启发式优化方法种类繁多，包括经典的模拟退火方法、遗传算法、蚁群算法以及粒子群算法等等。 |
| 拉格朗日乘数法 | 解决约束优化问题。 |

**11. 优化算法及其优缺点？**

|  |  |  |
| --- | --- | --- |
| 优化算法 | 优点 | 缺点 |
| 梯度下降 | ①收敛速度快  ②易于并行实现 | 容易陷入局部最优解 |
| 随机梯度下降 | 可以一定程度上结局局部最优解的问题 | 收敛速度慢 |
| 小批量梯度下降 | 综合随机梯度下降和批量梯度下降的优缺点，提出的一个中和的方法 | |
| 牛顿法 | 二阶收敛，收敛速度快 | 每一步都需要求解目标函数的海森矩阵的逆矩阵，计算比较复杂 |
| 拟牛顿法 | 克服了牛顿法海森矩阵求解复杂的问题 | |

**12. 梯度下降的步长怎么决定？有什么影响？**

①line search：基本思想就是每次试一个步长，如果用该步长走的话，看函数值会不会比当前点下降一定的程度，如果没有，就按比例减小步长，再试，直到满足条件（根据泰勒展开式我们知道步长足够小时总会满足下降条件）。所以line search实际上是计算量比较大的，不过在以前数据量不大的情况下这都不是问题。

②步长决定了在梯度下降迭代的过程中，每一步沿梯度负方向前进的长度。用上面下山的例子，步长就是在当前这一步所在位置沿着最陡峭最易下山的位置走的那一步的长度。步长会影响算法的收敛速度。

**13. 拟牛顿法和牛顿法区别，哪个收敛快？为什么？**

①区别：在牛顿法的迭代中，需要计算海塞矩阵的逆矩阵，这一计算比较复杂，拟牛顿法用一个n阶矩阵来近似代替海塞矩阵。

②这个问题个人认为是拟牛顿法，因为拟牛顿法计算更快，而且使用了海森矩阵的近似矩阵，并不会产生太大误差。

**14. 梯度下降是最优解还是局部解？如何让它收敛到全局最优解？**

一般是局部最优解。可以使用SGD、MBGD等方法来避免陷入局部最优；可以考虑加入退火算法或者遗传算法之类的思想，简单说就是在搜索过程中不但有基于梯度下降的方向，同时也融入少量的逆向搜索，最终设定一个收敛域即可。

**15. 为什么在一般问题里梯度下降比牛顿类算法更常用呢？**

因为对于规模比较大的问题，Hessian计算是非常耗时的；同时对于很多对精度需求不那么高的问题，梯度下降的收敛速度已经足够了。

**16. 梯度下降法和牛顿法的对比。**

①牛顿法并没有一定收敛的保证，如果在当前迭代点二阶近似不准确，或者，目标函数在这里本就是线性，二阶导数根本不存在，那么会导致牛顿法不收敛。而梯度下降加上line search是一定会往正确方向收敛的，但是速度会慢。

②梯度下降法用目标函数的一阶偏导、以负梯度方向作为搜索方向，只考虑目标函数在迭代点的局部性质；牛顿法同时考虑了目标函数的一、二阶偏导数，考虑了梯度变化趋势，因而能更合适的确定搜索方向加快收敛，但牛顿法也存在以下缺点：

a. 对目标函数有严格要求，必须有连续的一、二阶偏导数，海森矩阵必须正定；

b. 计算量大，除梯度外，还需计算二阶偏导矩阵及其逆矩阵。

③牛顿法要比梯度下降法更具有全局判断能力：

梯度法从初始点的领域开始判断，在局部进行下降，然后步步逼近极值，往往是走之字型的。牛顿法在二阶导数的作用下，从函数的凸性出发，直接搜索怎样到达极值点，也就是说在选择方向时，不仅考虑当前坡度是否够大，还会考虑你走了一步之后，坡度是否会变得更大。从收敛速度来看，梯度下降是线性收敛，牛顿法是超线性的，至少二阶收敛。