**最小二乘法**

最小二乘法的目标：求误差的最小平方和，对应有两种：线性和非线性。线性最小二乘的解是closed-form即，而非线性最小二乘没有closed-form，通常用迭代法求解。迭代法，即在每一步update未知量逐渐逼近解，可以用于各种各样的问题（包括最小二乘），比如求的不是误差的最小平方和而是最小立方和。

**梯度下降**是迭代法的一种，可以用于求解最小二乘问题（线性和非线性都可以）

求解 J（a）的最小值 ，随意选择一个a(1),计算其梯度向量 deltaJ(a(1)),下一个值a(2)由a(1)向下降最抖的方向移动一段距离，得到，即延梯度的负方向。a(k+1)= a(k)- lr(k)deltaJ(a(k))直到收敛 。关键是学习率的选择，可以把a（k+1）看作是一个关于Ir（k）,这样就可以通过求满足J(a(k+1))的最小的Ir（k）。

**高斯-牛顿法**是另一种经常用于求解非线性最小二乘的迭代法（一定程度上可视为标准非线性最小二乘求解方法）。

求解方程f(x)=0并不是所有的方程都有求根公式，或者求根公式很复杂，导致求解困难。利用牛顿法，可以迭代求解。原理是利用泰勒公式，在x0处展开，且展开到一阶，即f(x) = f(x0)+(x－x0)f'(x0)求解方程f(x)=0，即f(x0)+(x-x0)\*f'(x0)=0，求解x = x1=x0－f(x0)/f'(x0)，因为这是利用泰勒公式的一阶展开，f(x) = f(x0)+(x－x0)f'(x0)处并不是完全相等，而是近似相等，这里求得的x1并不能让f（x）=0，只能说f(x1)的值比f(x0)更接近f（x）=0，于是乎，迭代求解的想法就很自然了，可以进而推出x(n+1)=x(n)－f(x(n))/f'(x(n))，通过迭代，这个式子必然在f（x\*）=0的时候收敛。

牛顿法用于最优化,在最优化的问题中，线性最优化至少可以使用单纯行法求解，但对于非线性优化问题，牛顿法提供了一种求解的办法。假设任务是优化一个目标函数f，求函数f的极大极小问题，可以转化为求解函数f的导数f'=0的问题，这样求可以把优化问题看成方程求解问题（f'=0）。剩下的问题就和第一部分提到的牛顿法求解很相似了。

**比较**

牛顿法是二阶收敛，梯度下降是一阶收敛，所以牛顿法就更快。如果更通俗地说的话，比如你想找一条最短的路径走到一个盆地的最底部，梯度下降法每次只从你当前所处位置选一个坡度最大的方向走一步，牛顿法在选择方向时，不仅会考虑坡度是否够大，还会考虑你走了一步之后，坡度是否会变得更大。所以，可以说牛顿法比梯度下降法看得更远一点，能更快地走到最底部。

根据wiki上的解释，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。