**普通优化算法**

**优化算法的数学原理**

**AdaGrad(Adaptive Gradient)**

**步骤**

每次迭代时自适应地调整每个参数的学习率，在第𝑡次迭代时，先计算每个参数梯度平方的累计值

其中为按元素乘积， 为次迭代时的梯度

AdaGrad算法的参数更新差值为

其中是初始的学习率，是为了保持数值稳定性而设置的非常小的常数，这里的开平方、除、加运算都是按元素进行的操作

**核心思想**

在标准的梯度下降法中，每个参数在每次迭代时都使用相同的学习率。由于每个参数的维度上收敛速度都不相同，因此根据不同参数的收敛情况分别设置学习率：如果某个参数的偏导数累积比较大，则相对缩小其学习率；相反，如果其偏导数累积较小，则相对放大其学习率

**缺点**

对于任一参数，随着迭代次数的增加，学习率逐渐缩小，在经过一定次数的迭代依然没有找到最优点时，由 于这时的学习率已经非常小，很难再继续找到最优点

**RMSprop(Root Mean Square Propagation)**

**步骤**

首先计算每次迭代梯度平方的指数衰减移动平均

其中为衰减率，一般取值0.9

RMSprop算法的参数更新差值为

其中为初始学习率，可设为0.001

**核心思想**

可以在有些情况下避免 AdaGrad 算法中学习率不断单调下降以至 于过早衰减的缺点，在迭代过程中，每个参数的学习率并不是单调呈递减趋势，既可以变小也可以变大

**动量法**

**数学模型**

其中𝜌为动量因子，通常设为0.9，𝛼为学习率

**核心思想**

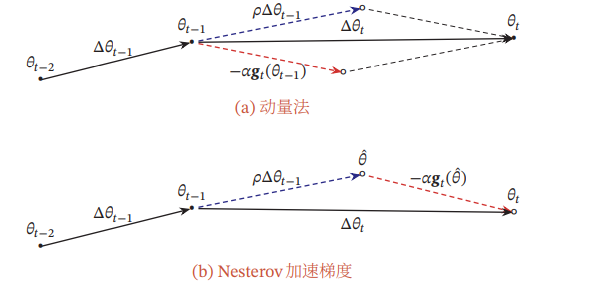
用之前积累动量来替代真正的梯度．每次迭代的梯度可以看作加速度

**Nesterov加速梯度(Nesterov Accelerated Gradient)**

**数学模型**

表示损失函数在点上的偏导数

**核心思想**



在动量法中，可以被拆分为两步进行，先根据更新一次得到参数，再用进行更新

其中梯度为点上的梯度，因此在第二步更新中有些不太合理．更合理的更新方向应该为上的梯度．

**Adam(Adaptive Moment Estimation Algorithm)**

**步骤**

一阶矩

二阶矩

偏差修正

通常取，

Adam算法的参数更新差值为

学习率通常设为0.001，并且也可以进行衰减，如

**核心思想**

Adam算法可以看作动量法和 RMSprop 算法的结合，不但使用动量作为参数更新方向，而且可以自适应调整学习率。Adam算法一方面计算梯度的指数加权平均(和动量法类似)，另一方面计算梯度平方的指数加权平均(与RMSprop算法类似）

那么在迭代初期和的值会比真实的一、二阶矩要小．特别是当和都接近于1时，偏差会很大。因此，需要对偏差进行修正