梯度下降

2020年5月10日 星期日 21:10

梯度下降是机器学习中最常用的优化方法，主要分两类

1. 批量梯度下降 BGD

例如现在有一个需要拟合的函数 $h=\sum_{j=0}^{n}\theta_j x_j$ 其实跟 $h=wx$ 是一个意思

这时我们可以计算出数据集中每个数据的误差并加和作为 loss

$L(\theta)=\frac{1}{m}\sum_{i=1}^{m}(y_i-h)$

如此一来，我们可以求 $L(\theta)$ 对于每一个 θ 的梯度为 $\frac{dL(\theta)}{\theta_j}$

此时就用相应的梯度去修改相应的 θ

$\theta_j'=\theta_j-\lambda\frac{dL(\theta)}{d\theta_j}$ λ 为学习率

这里其实可以看出来每修改一个权值都要动用所有的数据，而且每一次依照学习率只要向着梯度方向优化一小步，每优化一小步都要所有数据，这样太慢

除了速度慢以外，一旦在优化过程中 遇到鞍点、或较差的局部最小值，就会停止更新

2. 随机梯度下降

SGD可以解决上面两种问题

前面用所有数据的和作为loss，而SGD的loss则只针对一个数据：

$L(\theta)=y_i-h$

这时在优化的过程中也是发现一个分类错误的点就更新一次

$\theta_j'=\theta_j-\lambda\frac{L(\theta)}{d\theta_j}$

由于是每发现一个错误点，就更新一次，所以其实在优化过程中包含一定的随机性，会包含很多噪声，

但它比较快，而且在噪声不那么离谱的情况下效果也非常不错，最终可以落在全局最优附近

它本身有更大的机率逃离鞍点，但依然比较有限（从某种意义上说 Resnet 的意义就在于让SGD更容易找到全局最优）