学习率与最优化方法

目录

- ◆ 最优化概述
- ◆ 常见最优化方法

最优化概述

最优化概述

- 最优化是应用数学的分支，主要研究在特定情况下最大化或最小化函数或变量

  - 已知标量y和向量X，要确定一个函数（ ），使得 （（ ）） 尽量小

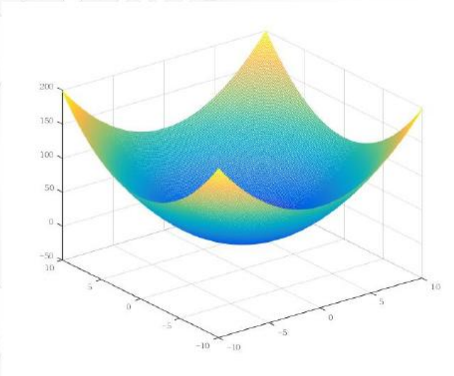
    {（）（）} ,（） （ （） （） （））

  - 常见的损失函数 （（ ）） （ ）

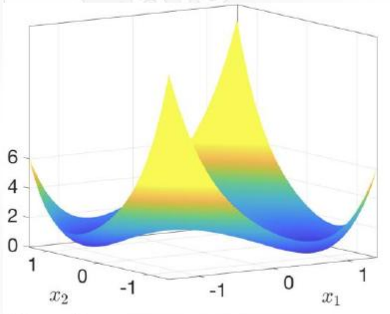
    （（ ）） （（ ））

凸优化与非凸优化目标

◆ 优化目标有**凸函数和非凸函数**两种



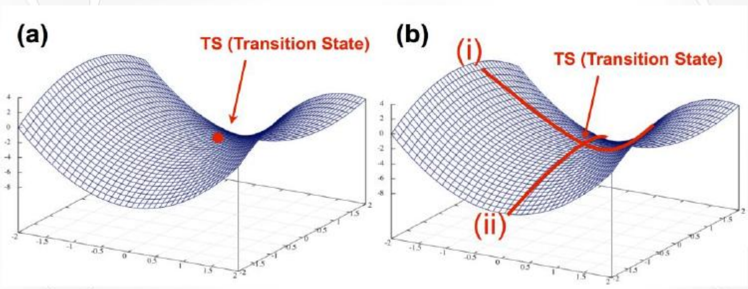
全局最小值=局部最小值

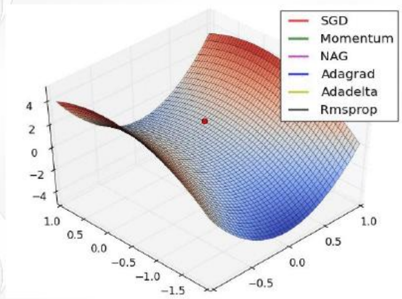


包含很多局部最小值

极值点与鞍点

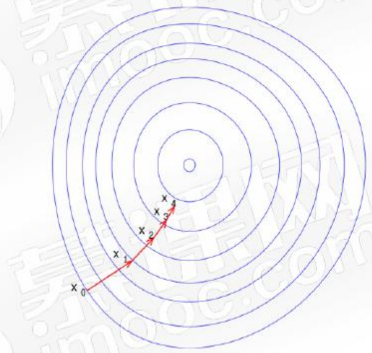
◆ 低维空间中，局部极小值很常见。高维空间中，**鞍点**（横截面上的局部极小值，某一些方向梯度下降，另一些方向梯度上升）更加常见





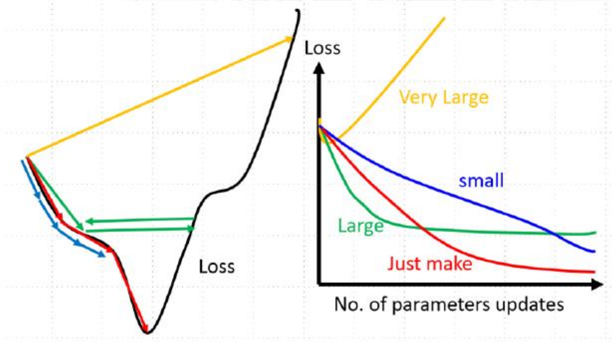
梯度下降算法与学习率

◆ 学习率是梯度下降算法中参数更新的步长(乘因子)



函数 , 为梯度，对于足够小的正数，有
(( )) ( )

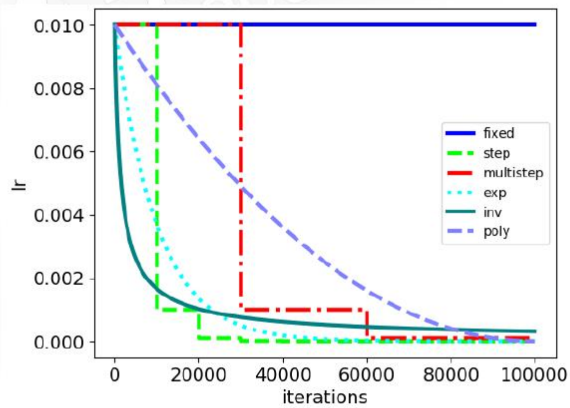
LR *



学习率对优化的影响

学习率迭代策略

◆ 学习率在训练过程中不会一直保持不变



常见的学习率衰减（learning rate decay）策略

最优化方法分类

最优化方法分类

◆ 常用的一阶优化方法

- 随机梯度下降SGD法
- momentum动量法
- Nesterov accelerated gradient法

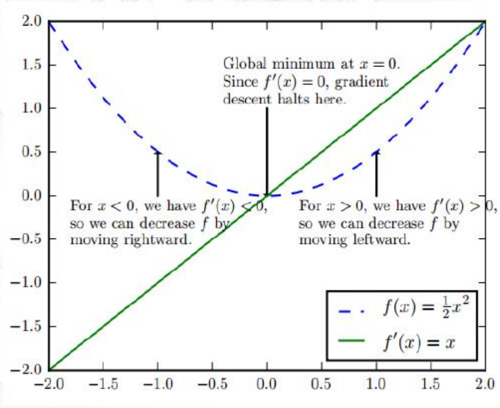
→ 基于选择更好的更新**方向**

- Adagrad法
- Adadelta与Rmsprop法
- Adam，AdaMax，Nadam法

→ 基于选择更为合理的**学习率**

随机梯度下降算法

◆ Stochastic gradient descent（SGD），沿着梯度反方向进行更新



沿梯度方向---->函数值上升；
沿梯度的反方向---->函数值下降；

参数更新 ( )

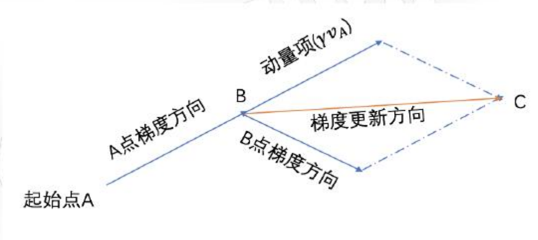
优点：简单
缺点：不稳定，学习率敏感，迭代慢

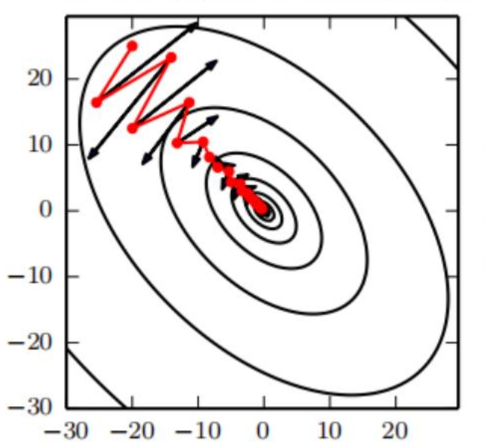
动量法Momentum

◆ 加速的SGD方法，积累了之前梯度指数级衰减的移动平均，并且继续沿该方向移动。

( )

默认 =0.9



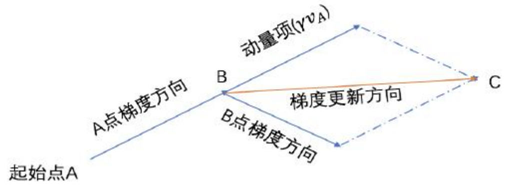


如果梯度方向不变，就越发更新的快，反之减弱，当前保证梯度收敛。

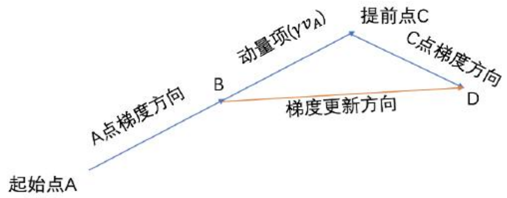
NAG法

◆ Nesterov accelerated gradient，在动量法中添加了一个校正因子

动量法 ( )



NAG法



要求梯度下降更快，更加智能，直接先按照前一次梯度方向更新一步将它作为当前的梯度

Adagrad法

◆ 自适应地为各个维度的参数分配不同的学习率

是当前的梯度， 是初始学习率， 是梯度平方累计值， 是一个比较小的数。

$\sqrt{}$

- 优点： 较小的时候，能够放大梯度， 较大的时候，能够约束梯度（激励+惩罚）
- 缺点：梯度累积导致学习率单调递减，后期学习率非常小，需要设置一个合适的全局初始学习率

Adadelta与RMSprop法

◆ 与Adagrad不同，只累加了一个窗口的梯度，使用动量平均计算

- 优点：保留了Adagrad调节不同维度学习率的优势
- 缺点：训练后期反复在局部最小值附近抖动。

Adam算法

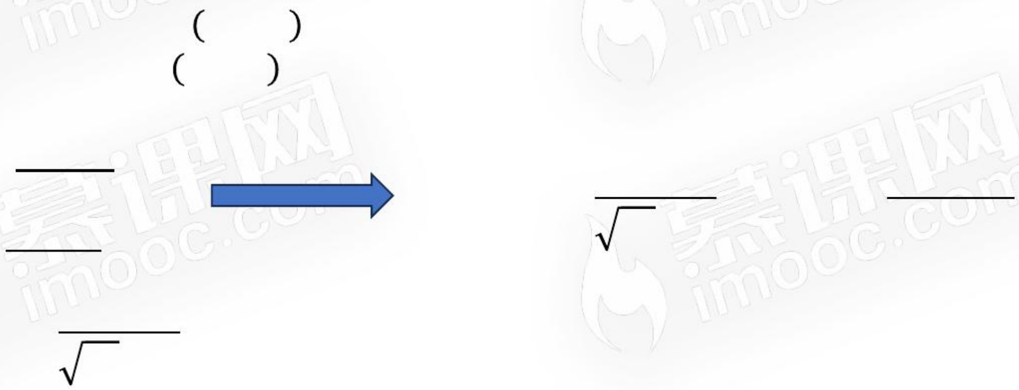
◆ 对梯度的一阶和二阶都进行了估计与偏差修正，使用梯度的一阶矩估计和二阶矩估计来动态调整每个参数的学习率

- 优点：对学习率没有那么敏感，学习步长有一个确定的范围，参数更新比较稳。
- 缺点：学习率在训练的后期仍然可能不稳定导致无法收敛到足够好的值，泛化能力较差。

将Adam使用的二阶矩变成更高阶，就成了Adamax算法。

Nadam算法

- NAG加上Adam，就成了Nadam方法，即带有动量项的Adam 。



二阶优化方法为何不用?

◆ 优点：二阶的方法因为使用了导数的二阶信息，因此其优化方向更加准确，速度也更快

◆ 缺点：二阶方法通常需要直接计算或者近似估计Hessian 矩阵，一阶方法一次迭代更新复杂度为O(N)，二阶方法就是O(N*N)，计算量大

下次预告：数据预处理工程