常见优化算法优劣对比

20195732 李御霖

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0.1 | 0.01 | 0.001 | 0.0001 | 0.00001 |
| sgd | 0.8687 | 0.1135 | 0.1135 | 0.1135 | 0.1135 |
| sgd+momentum | 0.9868 | 0.2406 | 0.1135 | 0.1135 | 0.1135 |
| adam | 0.0982 | 0.988 | 0.987 | 0.9256 | 0.2295 |

先看结论，表格内数据为test数据集的预测准确率,momentum为0.9

背景：

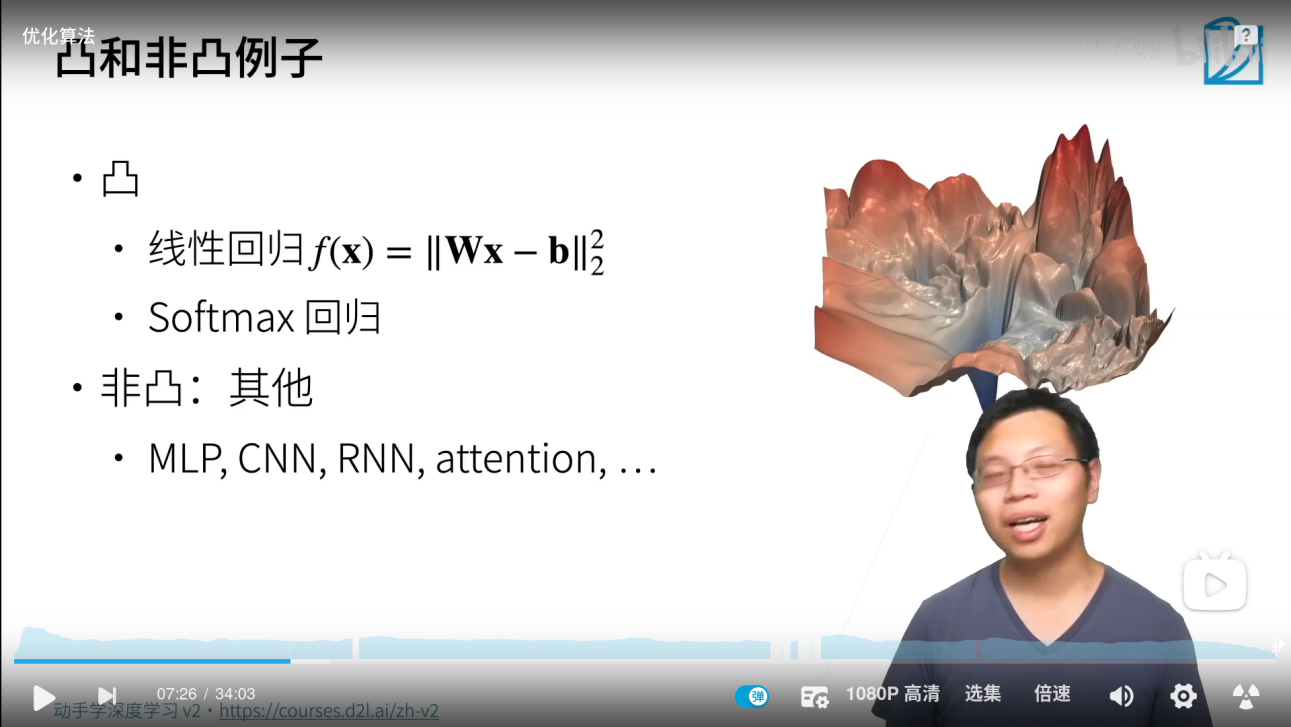
最优化问题:寻找函数f(x)的最小值，其中x∈C(C为制定集合，比如所有的x需要大于0)。

凸函数：函数f是凸当且仅当：f(ax+(1-a)y)<=af(x)+(1-a)f(y)，a属于(0,1),x!=y，且此时是严格凸函数。说白了就是函数上任意两点（x,y）连线，x,y之间的函数值都小于f(x),f(y)

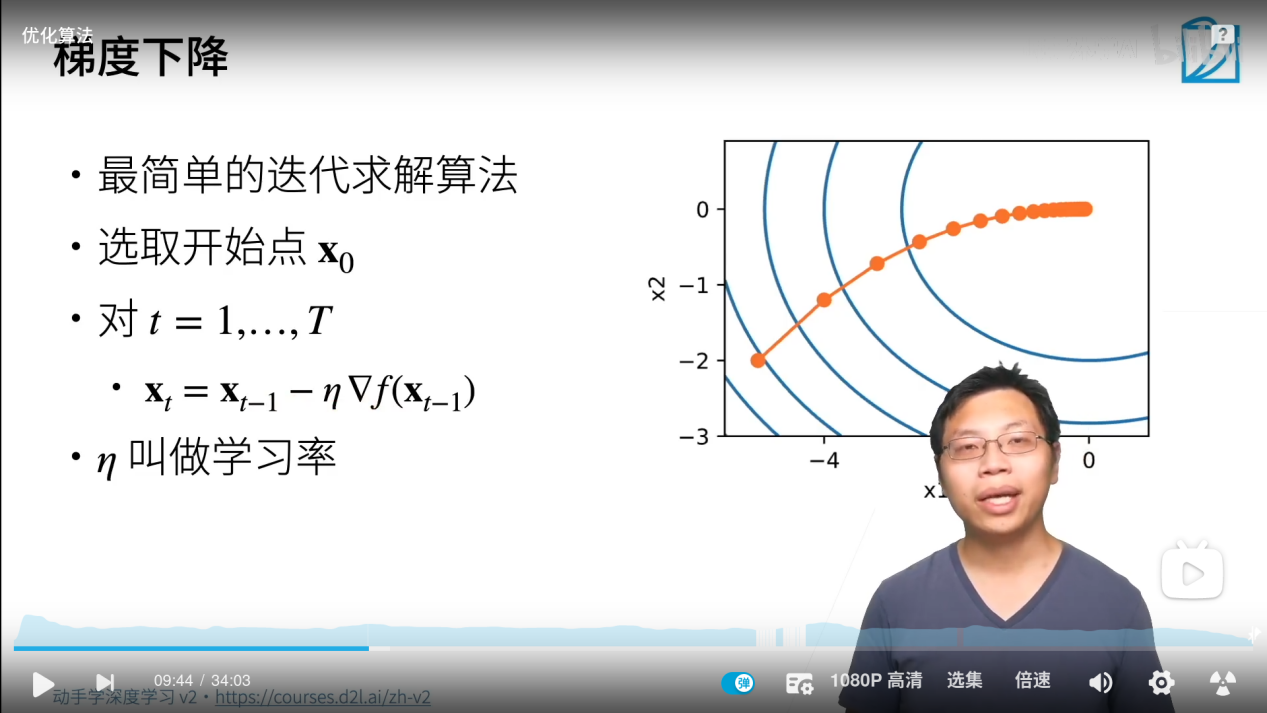
这个时候往往只有一个最值点。

神经网路中的‘‘f(x)’’是什么：其实就是全部的前向推理过程，即loss(net(x)),其中x是输入值。最优化的任务也是让损失函数最小。从而达成训练的目的。

不幸的事情：在神经网络中绝大多数的网络都是非凸的，凸函数我们常接触的只有两个：线性回归以及sotfmax回归。其他的多层感知机，卷积等由于各式各样的激活函数加入函数已经非凸。可以想象成凹凸不平的山谷，而我们的任务是下山。

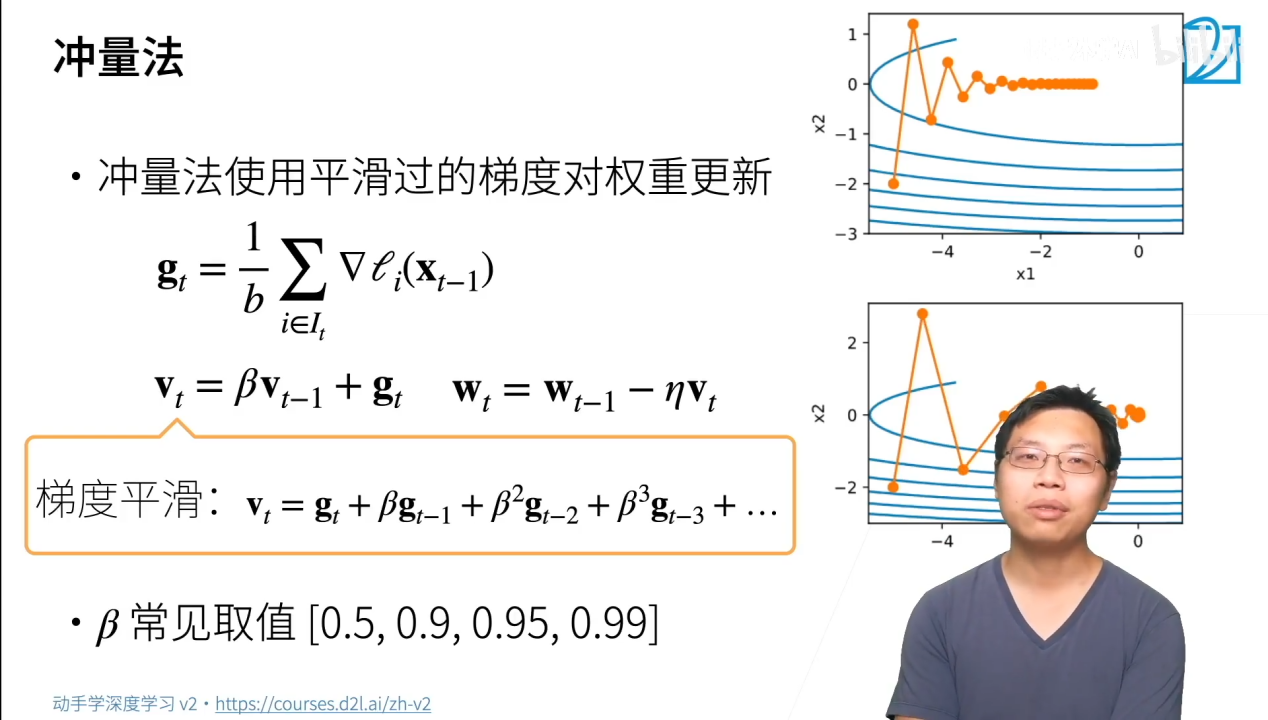
\*（复杂函数可视化）

**算法原理：**

**SGD：**

简单来讲就是每次计算后，记录平均梯度之后再反向传播。简单粗暴。

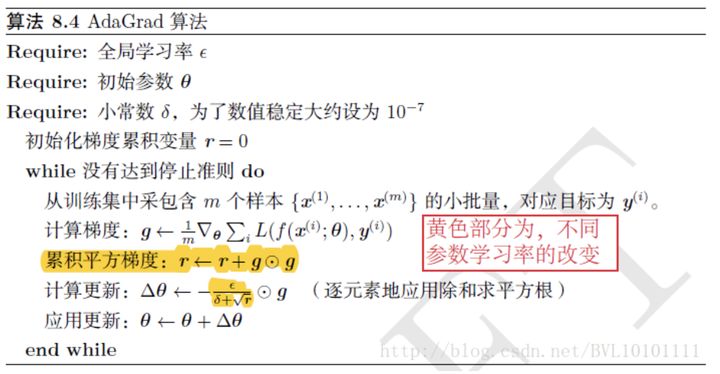
**Momentum：**



简单来说就是利用历史的梯度信息，保证梯度变化率不会过大，减少了点停留在局部最优解的可能性。

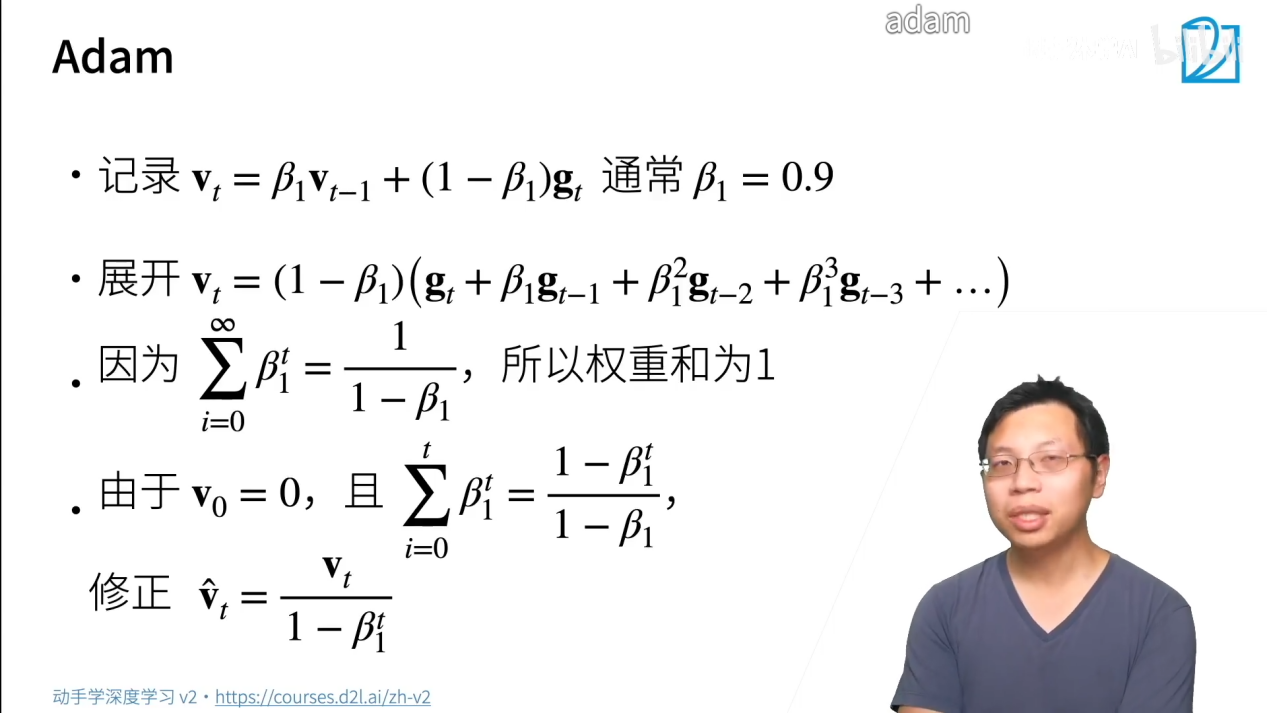
这里滤波的对象是梯度。

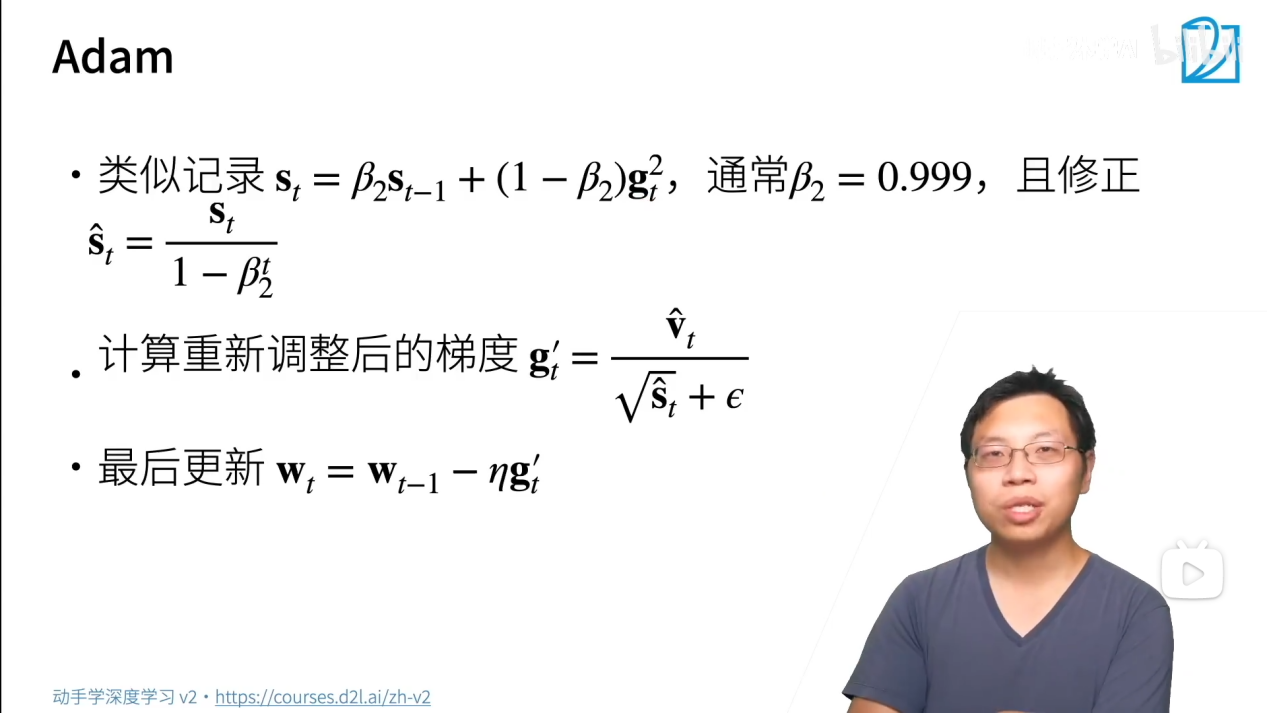
AdaGrade：



简单来讲，设置全局学习率之后，每次通过，全局学习率逐参数的除以历史梯度平方和的平方根，使得每个参数的学习率不同。但是这里由于梯度一直累积，往往会导致最后变化率接近0，实际中不建议使用。

**Adam:**

****

****

从完整公式中可以看出，最终的表达式里， v hat t是动量法的部分，利用调和平均对历史的梯度做记录进行平滑。下半部分，的s是对每个元素的平方进行记录，之后再进行平滑。复杂的公式也看不懂，结论上看就是同时对参数和梯度进行了平滑，并且对学习率不敏感。也就是说可以少调参了。

**实际经验和优劣：**

**SGD：**

优点：尽管看起来很粗糙，但是核心问题在于，一堆又一堆的平滑会不会让函数错过更优的解？经验上来看是的，往往只有细致的调整配合sgd才能让精度不断的上升，所以往往是先使用动量法或者adam进行余弦学习率迭代。在没法继续提升的时候，再上sgd进行调参，说不定运气好就到最优解了呢？

缺点：不收敛，容易卡局部最优。如果只使用sgd方法。很容易遇到不收敛，梯度爆炸或者消失，以及卡最局部解的情况。经常需要保存参数后更改学习率重新计算，很是麻烦

动量法：

优点：调用方便，往往和sgd绑定在一起，加少量参数就可以得到不错的效果。

缺点：没有对参数进行平滑，还是容易出现和sgd一样的问题。（比如在局部最优的时候就卡住）

AdaGrad：

因为没有使用经验，只能根据公式猜测。

优点：应该和adam类似，收敛很快

缺点：但是如果迭代过久肯定是找不到很好的解的，loss会在一堆值里波动

Adam：

优点：收敛快，不怎么需要调参。实际用起来爽歪歪。总是能找到一个不错的解。

缺点：因为对学习率不敏感，所以其实会发现adam最后会不停的在一个范围浮动。这个时候还是得靠sgd让他稳定下来。

再次看实际实验：

实验表格如下：

这里采用20个epoch后的测试数据集作为精度：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0.1 | 0.01 | 0.001 | 0.0001 | 0.00001 |
| sgd | 0.8687 | 0.1135 | 0.1135 | 0.1135 | 0.1135 |
| sgd+momentum | 0.9868 | 0.2406 | 0.1135 | 0.1135 | 0.1135 |
| adam | 0.0982 | 0.988 | 0.987 | 0.9256 | 0.2295 |

实验结论和前面的理论分析是一致的。唯一有一个问题是adam在0.1的时候不收敛了，或者说结果很奇怪？

由于时间限制，这里个人认为是学习率过大，数值不断跳变接近随机，准确率接近0.1。