# 一个框架看懂优化算法之异同 SGD/AdaGrad/Adam


Juliuszh
清华大学 计算机科学与技术博士

取消关注



## Adam那么棒，为什么还对SGD念念不忘 (1) —— 一个框架看懂优化算法

机器学习界有一群炼丹师，他们每天的日常是：

拿来药材（数据），架起八卦炉（模型），点着六味真火（优化算法），就摇着蒲扇等着丹药出炉了。

不过，当过厨子的都知道，同样的食材，同样的菜谱，但火候不一样了，这出来的口味可是千差万别。火小了夹生，火大了易糊，火不匀则半生半糊。

机器学习也是一样，模型优化算法的选择直接关系到最终模型的性能。有时候效果不好，未必是特征的问题或者模型设计的问题，很可能就是优化算法的问题。

说到优化算法，入门级必从SGD学起，老司机则会告诉你更好的还有AdaGrad/AdaDelta，或者直接无脑用Adam。可是看看学术界的最新paper，却发现一众大神还在用着入门级的SGD，最多加个Moment或者Nesterov，还经常会黑一下Adam。比如 UC Berkeley的一篇论文就在Conclusion中写道：

> Despite the fact that our experimental evidence demonstrates that adaptive methods are not advantageous for machine learning, the Adam algorithm remains incredibly popular. We are not sure exactly as to why ……

无奈与酸楚之情溢于言表。

这是为什么呢？难道平平淡淡才是真？

首先我们来回顾一下各类优化算法。

深度学习优化算法经历了 SGD -> SGDM -> NAG ->AdaGrad -> AdaDelta -> Adam -> Nadam 这样的发展历程。Google一下就可以看到很多的教程文章，详细告诉你这些算法是如何一步一步演变而来的。在这里，我们换一个思路，用一个框架来梳理所有的优化算法，做一个更加高屋建瓴的对比。

首先定义：待优化参数： $w$ ，目标函数： $f(w)$ ，初始学习率 $\alpha$ 。

而后，开始进行迭代优化。在每个epoch $t$ ：

1. 计算目标函数关于当前参数的梯度： $g_t = \nabla f(w_t)$
2. 根据历史梯度计算一阶动量和二阶动量： $m_t = \phi(g_1, g_2, \cdots, g_t); V_t = \psi(g_1, g_2, \cdots, g_t)$ ，
3. 计算当前时刻的下降梯度： $\eta_t = \alpha \cdot m_t / \sqrt{V_t}$
4. 根据下降梯度进行更新： $w_{t+1} = w_t - \eta_t$

掌握了这个框架，你可以轻轻松松设计自己的优化算法。

我们拿着这个框架，来照一照各种玄乎其玄的优化算法的真身。步骤3、4对于各个算法都是一致的，主要的差别就体现在1和2上。

## SGD

先来看SGD。SGD没有动量的概念，也就是说：

$$m_t = g_t; V_t = I^2$$

代入步骤3，可以看到下降梯度就是最简单的

$$\eta_t = \alpha \cdot g_t$$

SGD最大的缺点是下降速度慢，而且可能会在沟壑的两边持续震荡，停留在一个局部最优点。

为了抑制SGD的震荡，SGDM认为梯度下降过程可以加入惯性。下坡的时候，如果发现是陡坡，那就利用惯性跑的快一些。SGDM全称是SGD with momentum，在SGD基础上引入了一阶动量：

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$$

一阶动量是各个时刻梯度方向的指数移动平均值，约等于最近 $1/(1-\beta_1)$ 个时刻的梯度向量和的平均值。

也就是说，t时刻的下降方向，不仅由当前点的梯度方向决定，而且由此前累积的下降方向决定。$\beta_1$ 的经验值为0.9，这就意味着下降方向主要是此前累积的下降方向，并略微偏向当前时刻的下降方向。想象高速公路上汽车转弯，在高速向前的同时略微偏向，急转弯可是要出事的。

## SGD with Nesterov Acceleration

SGD 还有一个问题是困在局部最优的沟壑里面震荡。想象一下你走到一个盆地，四周都是略高的小山，你觉得没有下坡的方向，那就只能待在这里了。可是如果你爬上高地，就会发现外面的世界还很广阔。因此，我们不能停留在当前位置去观察未来的方向，而要向前一步、多看一步、看远一些。

NAG全称Nesterov Accelerated Gradient，是在SGD、SGD-M的基础上的进一步改进，改进点在于步骤1。我们知道在时刻t的主要下降方向是由累积动量决定的，自己的梯度方向说了也不算，那与其看当前梯度方向，不如先看看如果跟着累积动量走了一步，那个时候再怎么走。因此，NAG在步骤1，不计算当前位置的梯度方向，而是计算如果按照累积动量走了一步，那个时候的下降方向：

$$g_t = \nabla f(w_t - \alpha \cdot m_{t-1}/\sqrt{V_{t-1}})$$

然后用下一个点的梯度方向，与历史累积动量相结合，计算步骤2中当前时刻的累积动量。

## AdaGrad

此前我们都没有用到二阶动量。二阶动量的出现，才意味着“自适应学习率”优化算法时代的到来。SGD及其变种以同样的学习率更新每个参数，但深度神经网络往往包含大量的参数，这些参数

怎么样去度量历史更新频率呢？那就是二阶动量——该维度上，迄今为止所有梯度值的平方和：

$$V_t = \sum_{\tau=1}^{t} g_\tau^2$$

我们再回顾一下步骤3中的下降梯度：

$$\eta_t = \alpha \cdot m_t / \sqrt{V_t}$$

可以看出，此时实质上的学习率由 $\alpha$ 变成了 $\alpha/\sqrt{V_t}$ 。一般为了避免分母为0，会在分母上加一个小的平滑项。因此 $\sqrt{V_t}$ 是恒大于0的，而且参数更新越频繁，二阶动量越大，学习率就越小。

这一方法在稀疏数据场景下表现非常好。但也存在一些问题：因为 $\sqrt{V_t}$ 是单调递增的，会使得学习率单调递减至0，可能会使得训练过程提前结束，即便后续还有数据也无法学到必要的知识。

## AdaDelta / RMSProp

由于AdaGrad单调递减的学习率变化过于激进，我们考虑一个改变二阶动量计算方法的策略：不累积全部历史梯度，而只关注过去一段时间窗口的下降梯度。这也就是AdaDelta名称中Delta的来历。

修改的思路很简单。前面我们讲到，指数移动平均值大约就是过去一段时间的平均值，因此我们用这一方法来计算二阶累积动量：

$$V_t = \beta_2 * V_{t-1} + (1 - \beta_2) g_t^2$$

这就避免了二阶动量持续累积、导致训练过程提前结束的问题了。

## Adam

阶动量和二阶动量都用起来，就是Adam了——Adaptive + Momentum。

SGD的一阶动量：

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$$

加上AdaDelta的二阶动量：

$$V_t = \beta_2 * V_{t-1} + (1 - \beta_2) g_t^2$$

优化算法里最常见的两个超参数 $\beta_1, \beta_2$ 就都在这里了，前者控制一阶动量，后者控制二阶动量。

## Nadam

最后是Nadam。我们说Adam是集大成者，但它居然遗漏了Nesterov，这还能忍？必须给它加上，按照NAG的步骤1：

$$g_t = \nabla f(w_t - \alpha \cdot m_{t-1} / \sqrt{V_t})$$

这就是Nesterov + Adam = Nadam了。

说到这里，大概可以理解为什么j经常有人说 Adam / Nadam 目前最主流、最好用的优化算法了。新手上路，先拿来一试，收敛速度嗖嗖滴，效果也是杠杠滴。

那为什么Adam还老招人黑，被学术界一顿鄙夷？难道只是为了发paper灌水吗？

请继续阅读：

Adam那么棒，为什么还对SGD念念不忘 (2)—— Adam的两宗罪

Adam那么棒，为什么还对SGD念念不忘 (3)—— 优化算法的选择与使用策略

前面我们讲到，一阶动量和二阶动量都是按照指数移动平均值进行计算的：

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$$

$$V_t = \beta_2 \cdot V_{t-1} + (1 - \beta_2) \cdot g_t^2$$

实际使用过程中，参数的经验值是

$$\beta_1 = 0.9, \beta_2 = 0.999$$

初始化：

$$m_0 = 0, V_0 = 0$$

这个时候我们看到，在初期， $m_t, V_t$ 都会接近于0，这个估计是有问题的。因此我们常常根据下式进行误差修正：

$$\tilde{m}_t = m_t / (1 - \beta_1^t)$$

$$\tilde{V}_t = V_t / (1 - \beta_2^t)$$

————————————————————

行有所思，学有所得，陋鄙之言，请多指教。

欢迎关注我的微信公众号 Julius-AI

编辑于 2018-09-12

「真诚赞赏，手留余香」

赞赏

7 人已赞赏