狂徒归来
人生如逆旅，我亦是行人

深度学习中常用的优化器简介

深度学习中常用的优化器简介

SGD

mini-batch SGD 是最基础的优化方法，是后续改良方法的基础。下式给出SGD的更新公式

$$\theta_t = \theta_{t-1} - \alpha \nabla_\theta J(\theta)$$

其中α是学习速率。

SGD with Momentum

带动量的mini-SGD的更新方法如下

$$v_t = r \cdot v_{t-1} + \alpha \nabla_\theta J(\theta)$$
$$\theta_t = \theta_{t-1} - v_t$$

如果这一次的梯度与上一次的梯度方向一致，那么更新量就会越来越大，这样沿着负梯度的方向就会越走越快，可以使得模型收敛加速。

Nesterov Momentum

Nesterov Momentum 是 SGD with Momentum 的改进版，应用该方法的参数更新策略如下

$$v_t = r \cdot v_{t-1} + \alpha \cdot \nabla_\theta J(\theta - r \cdot v_{t-1})$$
$$\theta_t = \theta_{t-1} - v_t$$

在计算梯度的时候，加入了预估的信息，这样可以在上坡之前提前减速，减少震荡，使得优化朝着更加有利的方向进行。

Adagrad

前面介绍的三种方法，所有的参数使用着完全一样的学习速率。但是，讲道理，不同的参数应该使用不同的学习速率，比如出现频率较低的参数更新幅度应该大，而频率高的参数更新幅度就相对小一些。AdaGrad正是这样的方法，更为具体的

$$v_t = v_{t-1} + g_t^2$$
$$\theta_t = \theta_{t-1} - \frac{\alpha}{\sqrt{v_t + \epsilon}} \cdot g_t$$

其中ϵ是平滑因子，避免被开方的数是0。这里解释了为什么更新频率低的参数其更新量相对会大些，因为这些参数对应的分母较小。但是，AdaGrad优化器也有着明显的缺点，当v_t累积到足够的大的时候，分式的结果会无限接近0，导致参数更新缓慢甚至根本无法被更新，使得训练提前结束。

RMSprop

RMSProp是AdaGrad的一种改良，其计算如下式所示：

2.leetcode 214. 最短回文串 解题报告
3.leetcode 211. 添加与搜索单词 - …
4.leetcode 149. 直线上最多的点数 …
5.leetcode 208. 实现 Trie (前缀树)
6.leetcode 201. 数字范围按位与 解…
7.leetcode 179. 最大数 解题报告
8.Python 装饰器初探
9.leetcode 174. 地下城游戏 解题报告
10.TensorFlow dataset API 使用

随笔分类 (1067)

2-SAT(7)
C/C++ compiler(1)
Cocos2d-js游戏开发(2)
KMP/AC自动机/后缀数组/后缀…
Python(3)
RMQ(7)
并查集(4)
动态规划(210)
分治/二分(23)
机器学习(30)

最新评论

1. Re:爬淘宝的商品信息下 （下） --实现定时任务爬取

@scheduler.scheduled_job('cron', hour=3, minute=0, id="daily_crawl") id 是?

--Blue·Sky

2. Re:学渣笔记之矩阵的 与迹

@ 冬之晓应该是...

--flyo

$$v_t = 0.9v_{t-1} + 0.1g_t^2$$
$$\theta_t = \theta_{t-1} - \frac{\alpha}{\sqrt{v_t + \epsilon}} g_t$$

可以看到，这里使用的是移动指数平均，不再是AdaGrad方法中的累加和，当β取0.9的时候，可以看作是最近10次梯度更新量的加权平均。

Adam

Adam是上述方法的集大成者，除了使用了梯度的平方移动加权均值，也使用了梯度本身的移动加权均值。其计算如下

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$$
$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$$
$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$$
$$\hat{v}_t = \frac{v_t}{1-\beta_2^t}$$
$$\theta_t = \theta_{t-1} - \frac{\alpha}{\sqrt{\hat{v}_t + \epsilon}} \hat{m}_t$$

从上面的表达式中，可以看到，在计算移动指数平均的时候，还进行了修正，避免了移动指数平均的冷启动问题。

> 夜空中最亮的星，照亮我前行

分类: 机器学习

标签: 优化器 , Adam , SGD , AdaGrad , RMSProp , Nesterov , Momentum

3. Re:拉格朗日乘数法解含不等式约束的最优化问题

博主，想问一下，ADMM的增广拉格朗日函数能不能同样引入松弛变量去处理不等式约束呢

--EdoHan

阅读排行榜

1. 学渣笔记之矩阵的导数与迹(19…
2. 拉格朗日乘数法解含不等式约…

评论排行榜

1. BNU 4346 Scout YYF I(6)
2. HDU 4309 Seikimatsu Occult Ton…

推荐排行榜

1. SVM 为什么要从原始问题变为…
2. 学渣笔记之矩阵的导数与迹(3)
3. Python 黑魔法（持续收录）(2)
4. Tensorflow 自适应学习速率(2)
5. HDU 4309 Seikimatsu Occult Ton…

« 上一篇：GloVe词分布式表示
» 下一篇：Backpropagation Through Time (BPTT) 梯度消失与梯度爆炸

posted @ 2018-11-07 12:57 狂徒归来 阅读(711) 评论

历史上的今天：
2018-11-07 GloVe词分布式表示
2015-11-07 HDU 4426 Palindromic Substring
2015-11-07 HDU 3376 Matrix Again
2015-11-07 HDU 4044 GeoDefense
2014-11-07 POJ 3155 Hard Life