

CSDN新首页上线啦，邀请你来立即体验！(http://blog.csdn.net/)

CSDN

博客 (//blog.csdn.net/)

下载 (//download.csdn.net/?ref=toolbar)

更多 ▾

立即体验

登录 (https://passport.csdn.net/mobile/login?ref=toolbar)

注册 (http://passport.csdn.net/account/mobile/register?ref=toolbar&action=mobileRegister)

深入了解梯度下降算法

原创 2015年12月13日 17:03:31

标签：梯度下降 (http://so.csdn.net/so/search/s.do?q=梯度下降&t=blog) / 优化 (http://so.csdn.net/so/search/s.do?q=优化&t=blog) / 步长 (http://so.csdn.net/so/search/s.do?q=步长&t=blog) / Wolfe条件 (http://so.csdn.net/so/search/s.do?q=Wolfe条件&t=blog) / 梯度方向 (http://so.csdn.net/so/search/s.do?q=梯度方向&t=blog)



AlexInML (http://blog.csdn.net/wangjian1204)

+ 关注

| | | | |
|----|----|----|-----------------------------------|
| 原创 | 粉丝 | 喜欢 | 未开通 |
| 37 | 32 | 0 | (https://github.com/wangjian1204) |

- 他的最新文章
- 更多文章
- (http://blog.csdn.net/wangjian1204)
- Conda虚拟环境 (http://blog.csdn.net/wangjian1204/article/details/78508949)
- 深度学习目标检测之RPN-based方法 (http://blog.csdn.net/wangjian1204/article/details/78172588)
- flask快速搭建tensorflow http服务 (http://blog.csdn.net/wangjian1204/article/details/76732337)
- 在Spark上进行两个大数据集的匹配 (http://blog.csdn.net/wangjian1204/article/details/74906887)
- hadoop命令OutOfMemoryError GC (http://blog.csdn.net/wangjian1204/article/details/71732171)

一、目标函数：

首先明确一下本文的符号使用：向量用粗体表示，标量用普通的字母表示，例如： \mathbf{x} 表示一个向量， x 表示一个标量。

梯度下降算法在优化理论中有着很重要的地位，凭借实现简单、解决最优化问题效果较好并且有很好的普适性等优点梯度下降算法在机器学习等领域具有广泛的应用。梯度下降算法通常用来解决无约束最小化问题：

$$\min_{\mathbf{x} \in R^n} f(\mathbf{x})$$

(注意到无约束最大化问题 $\max_{\mathbf{x} \in R^n} f(\mathbf{x})$ 可以通过简单的在目标函数前加负号而转变成无约束最小化问题 $\min_{\mathbf{x} \in R^n} -f(\mathbf{x})$ ，所以只需要讨论最小化问题就行)

梯度下降算法迭代更新变量 \mathbf{x} 的值来寻找目标函数的局部极小值，第k+1 轮迭代时变量的更新规则：

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \cdot (-\nabla f)$$

其中 \mathbf{x}_k 是第k轮迭代后变量的值， α 是更新步长， ∇f 是目标函数导数（梯度）。

二、问题列表：

要对梯度下降算法有比较全面的了解首先需要回答以下几个问题：

- 1、为什么负梯度方向就是目标函数减小最快的方向？
- 2、迭代过程中步长如何选择？

- 相关推荐
- 梯度训练算法 (http://blog.csdn.net/Allyli0022/article/details/53583935)
- 梯度下降法 (Gradient Descent) (http://blog.csdn.net/isMarvellous/article/details/51098768)
- 梯度下降原理及在线性回归、逻辑回归中的应用 (http://blog.csdn.net/Erli11/article/details/36205505)
- 梯度的意义及在机器学习中的应用 (http://blog.csdn.net/bearshng/article/details/19073657)

三、梯度方向：

梯度下降算法看起来很简单，直接对目标函数求导就可以了。但是肯定会有些同学有这样的疑问：为什么负梯度方向就是目标函数减小最快的方向？在此做简单的证明，更严格的证明可以查阅本文最后的参考资料。证明比较简单，不需要用到很高深的数学知识。

先从比较简单的二元函数开始， $z = f(x, y)$ ，如果 $f(x, y)$ 在 (x_0, y_0) 点可微，则

$$f(x_0 + \Delta x, y_0 + \Delta y) - f(x_0, y_0) = f'_x(x_0, y_0)\Delta x + f'_y(x_0, y_0)\Delta y + o(\sqrt{(\Delta x)^2 + (\Delta y)^2})$$

其中 Δx 和 Δy 分别是在 x 轴和 y 轴方向上移动的距离，也就是在迭代过程中变量的更新量； $f'_x(x_0, y_0)$ 和 $f'_y(x_0, y_0)$ 分别是函数对 x 和 y 求偏导的结果。相信学过微积分的同学对这个式子会有映像。

我们的目标是在函数空间中移动一定的距离 h （趋向于零），即 $h = \sqrt{(\Delta x)^2 + (\Delta y)^2}$ 固定，的条件下使函数 f 的值下降最多。

因为长度 h 是固定的，所以总是可以找到某个角度

$$\Delta x = h \cdot \cos \bar{\alpha}, \Delta y = h \cdot \cos \bar{\beta}$$

$$(\cos \bar{\alpha})^2 + (\cos \bar{\beta})^2 = 1, \cos \bar{\beta} = \sin \bar{\alpha}, \bar{\alpha} + \bar{\beta} = 90^\circ$$

替换掉 Δx 和 Δy

$$\begin{aligned} \lim_{h \rightarrow 0} f(x_0 + h \cos \bar{\alpha}, y_0 + h \cos \bar{\beta}) - f(x_0, y_0) &= f'_x(x_0, y_0)h \cos \bar{\alpha} + f'_y(x_0, y_0)h \cos \bar{\beta} \\ &= h[f'_x(x_0, y_0) \cos \bar{\alpha} + f'_y(x_0, y_0) \cos \bar{\beta}] = h[f'_x(x_0, y_0), f'_y(x_0, y_0)][\cos \bar{\alpha}, \cos \bar{\beta}]^\top \end{aligned}$$

由于 h 是固定长度的标量，为了使 $\lim_{h \rightarrow 0} f(x_0 + h \cos \bar{\alpha}, y_0 + h \cos \bar{\beta}) - f(x_0, y_0)$ 最小，只能通过改变 $\cos \bar{\alpha}$ 和 $\cos \bar{\beta}$ 。而 $\bar{\alpha} + \bar{\beta} = 90^\circ$ ，其实就只需要确定 $\bar{\alpha}$ 的值。我们知道

$$\nabla f = [f'_x(x_0, y_0), f'_y(x_0, y_0)]$$

令 $\mathbf{u} = [\cos \bar{\alpha}, \cos \bar{\beta}]$ ，所以

$$\lim_{h \rightarrow 0} f(x_0 + h \cos \bar{\alpha}, y_0 + h \cos \bar{\beta}) - f(x_0, y_0) = h \nabla f \cdot \mathbf{u}^\top$$

根据向量内积的定义，

$$\nabla f \cdot \mathbf{u}^\top = |\nabla f| \cdot |\mathbf{u}^\top| \cdot \cos \angle(\nabla f, \mathbf{u}^\top) = |\nabla f| \cdot \cos \angle(\nabla f, \mathbf{u}^\top)$$

即 ∇f 在向量 \mathbf{u} 方向上的投影。所以当 ∇f 和 \mathbf{u} 方向相同时， $|\nabla f| \cdot \cos \angle(\nabla f, \mathbf{u}^\top) = |\nabla f|$ ，函数增长最快； ∇f 和 \mathbf{u} 方向相反时， $|\nabla f| \cdot \cos \angle(\nabla f, \mathbf{u}^\top) = -|\nabla f|$ ，函数减小最快。



广告



相亲网



他的热门文章

Spark把RDD数据保存到一个单个文件中 (<http://blog.csdn.net/wangjian1204/article/details/52422204>)

12954

PCA和SVD区别和联系 (<http://blog.csdn.net/wangjian1204/article/details/50642732>)

12822

五个例子掌握theano.scan函数 (<http://blog.csdn.net/wangjian1204/article/details/50518591>)

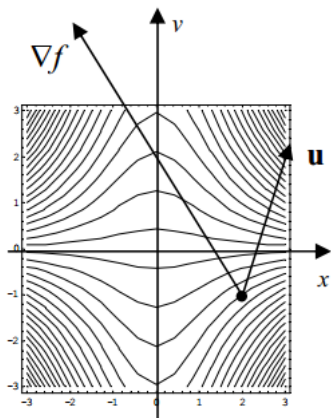
9264

Tensorflow Serving 模型部署和服务 (<http://blog.csdn.net/wangjian1204/article/details/68928656>)

8704

kmeans聚类算法及matlab实现 (<http://blog.csdn.net/wangjian1204/article/details/49803673>)

6237



二元函数的情况已经证完了，推广到多元的情况类似，这里就不再赘述了。

四、步长选择：

步长 α 的选择对梯度下降算法来说很重要， α 过小会导致收敛太慢； α 过大容易导致发散。下面介绍几种常用的选择 α 的方法，其中线性搜索方法理论基础较好，但是在实际中使用没有前两种那么广泛，究其原因主要还是因为前面两种方法使用简单并且在大多数情况下都可以得到满意的效果。

固定常数：

最简单的方法就是把 α 固定为一个常数，并且在迭代过程中不改变 α 的值。这种方法要求 α 较小，否则容易导致发散而无法收敛。

线性变化：

另一种简单的方法是在迭代的过程中不断的减小 α 的值。常用的赋值方法是 $\alpha = 1/k$ ，其中k是迭代的次数。也可以加入平滑因子 $\alpha = \tau / (k + \tau), \tau \in R$ 。

线性搜索（Linear Search）：

线性搜索算法把步长的选择看做一个优化问题：

$$\phi(\alpha) = f(\mathbf{x}_k + \alpha \mathbf{p}_k), \alpha > 0$$

其中 \mathbf{p}_k 是当前函数负梯度方向， $\phi(\alpha)$ 是步长 α 的函数，线性搜索方法利用函数 $\phi(\alpha)$ 找到一个合适的步长。首先介绍两个条件：Wolfe条件和Goldstein条件，它们是用来判断步长 α 是否足够好的准则。

有了一个合适的步长 α_k 之后，就可以通过下面的式子更新变量 \mathbf{x} 了：

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$$

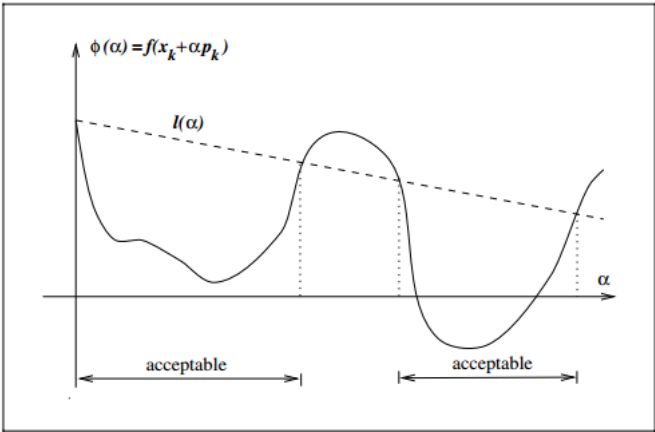
1、Wolfe 条件：

Wolfe条件主要包括两点：足够下降和曲率条件，在第k+1轮迭代时合适的步长 α_k 的值应该同时满足这两个条件。

f
1、足够下降条件要求 的函数值减小足够多

$$f(\mathbf{x}_k + \alpha_k \mathbf{p}_k) \leq f(\mathbf{x}_k) + c_1 \alpha_k \nabla f_k^\top \mathbf{p}_k, \quad c_1 \in (0, 1)$$

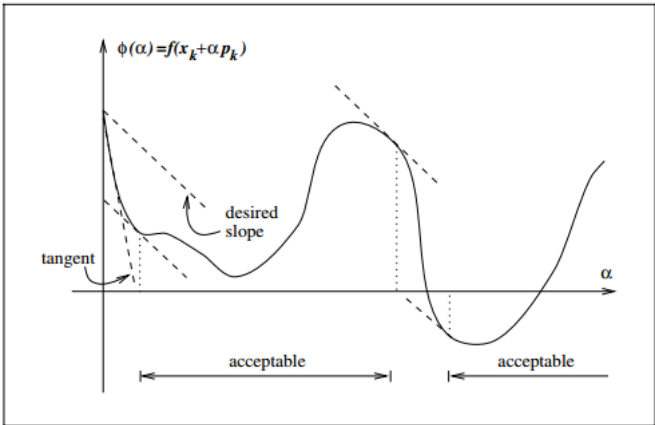
图中的 $l(\alpha)$ 就等于 $f(\mathbf{x}_k) + c_1 \alpha_k \nabla f_k^\top \mathbf{p}_k$ 。如果新的变量值 $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$ 使得函数值小于等于 $l(\alpha)$ 那么这个变量值是可以接受的 (acceptable) 。



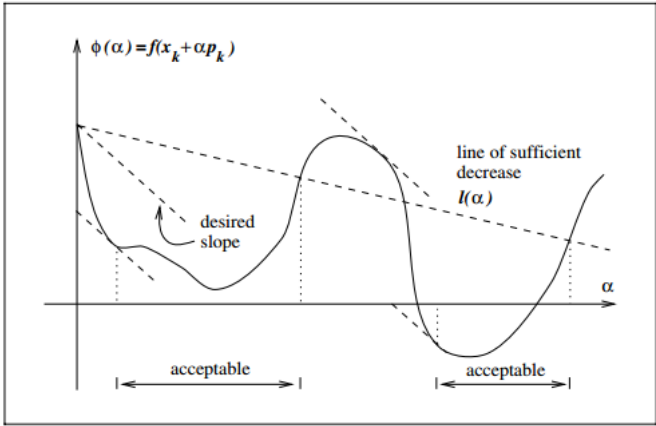
$\mathbf{x}_k + \alpha \mathbf{p}_k$ \mathbf{x}_k
2、曲率条件要求变量变化不能太小，即 不能与 太接近，否则收敛过慢

$$\nabla f(\mathbf{x}_k + \alpha_k \mathbf{p}_k)^\top \mathbf{p}_k \geq c_2 \nabla f_k^\top \mathbf{p}_k, \quad c_2 \in (c_1, 1)$$

图中的tangent虚线对应斜率 ∇f_k , 其他几条虚线对应斜率 $c_2 \nabla f_k$, 斜率在 ∇f_k 和 $c_2 \nabla f_k$ 之间的 $\nabla f(\mathbf{x}_k + \alpha_k \mathbf{p}_k)^\top \mathbf{p}_k$ 与 $\nabla f_k^\top \mathbf{p}_k$ 的内积 会大于 $\nabla f_k^\top \mathbf{p}_k$ (最快下降) , 小于 $c_2 \nabla f_k^\top \mathbf{p}_k$ 。所以满足曲率条件可以保证斜率 $\nabla f(\mathbf{x}_k + \alpha_k \mathbf{p}_k)$ 不在 ∇f_k 和 $c_2 \nabla f_k$ 之间。从而 \mathbf{x}_{k+1} 不会在 \mathbf{x}_k 附近 (从图中可以明显观察到) , 避免收敛过慢。



同时满足Wolfe两个条件的图如下，图中acceptable的区域表示满足Wolfe条件的区域：

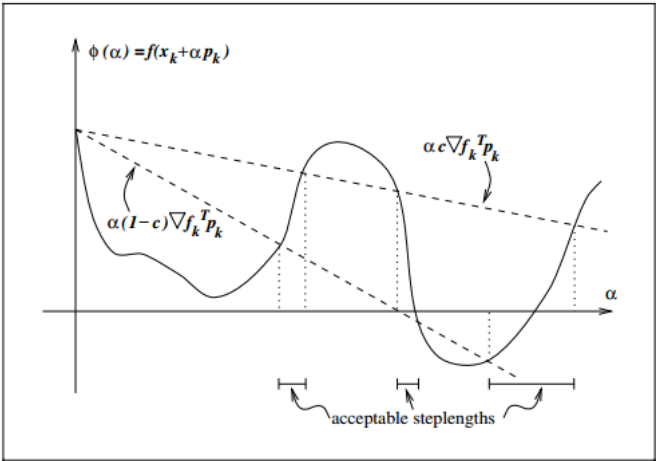


2、Goldstein 条件：

Goldstein条件和Wolfe很类似，Goldstein条件要求函数值有足够的下降，但是不要下降太多

$$f(\mathbf{x}_k) + (1 - c)\alpha_k \nabla f_k^\top \mathbf{p}_k \leq f(\mathbf{x}_k + \alpha_k \mathbf{p}_k) \leq f(\mathbf{x}_k) + c\alpha_k \nabla f_k^\top \mathbf{p}_k, \quad c \in (0, 1/2)$$

两条虚直线夹角之间的 α 值都满足Goldstein条件。



3、Backtracking Linear Search:

在实际应用中，大多数算法都不会完整使用Wolfe条件和Glodstein条件。Backtracking线性搜索是一种比较实用的步长选择算法，它只利用了Wolfe条件的足够下降条件。具体算法如图所示：

```
Choose  $\bar{\alpha} > 0, \rho \in (0, 1), c \in (0, 1)$ ; Set  $\alpha \leftarrow \bar{\alpha}$ ;  
repeat until  $f(x_k + \alpha p_k) \leq f(x_k) + c\alpha \nabla f_k^\top p_k$   
     $\alpha \leftarrow \rho \alpha$ ;  
end (repeat)  
Terminate with  $\alpha_k = \alpha$ .
```

Backtracking算法比较简单，它从一个较大的 α 值开始，不断按比例 ρ 缩小直到满足足够下降条件，最终得到 α_k 的值。



离婚女征婚



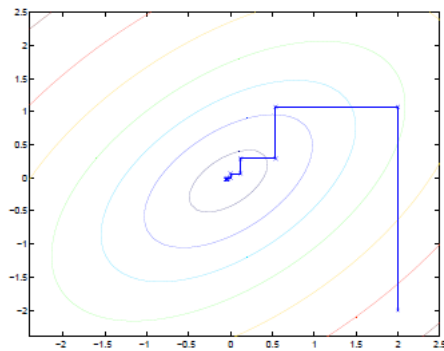
五、随机 (stochastic) 梯度下降和 Mini-batch梯度下降算法：



随机梯度下降算法和Mini-batch梯度下降算法都和原始的梯度下降算法类似，只是为了减少计算量，在一次迭代中只使用一个或者几个样本来更新变量 \mathbf{x} 。

六、坐标下降 (coordinate descent) 算法：

坐标下降算法和梯度下降算法比较类似：坐标下降算法在每一次迭代中在当前点处沿一个坐标方向进行一维搜索，固定其他的坐标方向，找到一个函数的局部极小值。在整个过程中依次循环使用不同的坐标方向进行迭代。



七、参考资料:

Numerical Optimization (Second Edition) by Jorge Nocedal, Stephen J. Wright.

版权声明：本文为博主原创文章，如需转载请在文章开头标明原始地址并联系博主。



内容举报



返回顶部



相关文章推荐

梯度训练算法 (<http://blog.csdn.net/Allyli0022/article/details/53583935>)

1. batch GD 每次迭代的梯度方向计算由所有训练样本共同投票决定，batch GD的损失函数是： $J(\theta) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x(i)) - y(i))^2$ 训练算法为：...

 Allyli0022 (<http://blog.csdn.net/Allyli0022>) 2016年12月12日 16:04  697

梯度下降法 (Gradient Descent) (http://blog.csdn.net/isMarvellous/article/details/5...

第一次写博客，好激动啊，哈哈。之前看了许多东西但经常是当时花了好大功夫懂了，但过一阵子却又忘了。现在终于决定追随大牛们的脚步，试着把学到的东西总结出来，一方面梳理思路，另一方面也作为备忘。接触机器学习...

 isMarvellous (http://blog.csdn.net/isMarvellous) 2016年04月08日 18:20  5742



票选结果：Python再上天，微软要求全员学Python？



宇宙语言Python荣登年度排行榜，吴恩达，微软纷纷为它站台，Python这么牛逼的原因是....

广告

(http://www.baidu.com/cb.php?c=Igf_pyfqHmknjnvPjc0IZ0qnfK9ujYzP1nYPH0k0Aw-5Hc3rHnYnHb0TAq15HfLPWRznjb0T1d-PHm3nWwBnhmLPjwWPyc30AwY5HDdnHnvrHnsnW60Igf_5y9YIZ0IQzq-uZR8mLPbUB48ugfEIAqspynETZ-YpAq8nWqdIAdxTvqdThP-5yF_UvTkn0KzujYk0AFV5H00TZcq0KdpyfqHRLPjnvnfKEpyfqHc4rj6kP0KWpyfqP1civrHnz0AqLUWYs0ZK45HcsP6KWThnqnHT3rj6)


梯度下降原理及在线性回归、逻辑回归中的应用 (http://blog.csdn.net/Erli11/article/detai...

参考文献：http://blog.sina.com.cn/s/blog_62339a2401015jyq.html

 Erli11 (http://blog.csdn.net/Erli11) 2014年07月03日 16:47  7113

梯度的意义及在机器学习中的应用 (http://blog.csdn.net/bearshng/article/details/19073...

今天一位考研的同学问及我梯度的概念，以及为什么在二元函数 $z=f(x,y)$ 明明表示一个三维空间曲面，为何其梯度是二维的。说实在话，至于为什么是二维的，当时我真的不清楚 一、梯度的概念 ...

 bearshng (http://blog.csdn.net/bearshng) 2014年02月11日 12:27  1171

梯度下降法 (http://blog.csdn.net/woxincd/article/details/7040944)

回归(regression)、梯度下降(gradient descent) 发表于332 天前 / 技术, 科研 / 评论数 3 / 被围观 1152 次+ 本文由Lef...

 woxincd (http://blog.csdn.net/woxincd) 2011年12月05日 09:33  151081



乳晕大是因为什么



牙科价目表



种头发危害



牙齿有洞怎么办



拔一颗智齿要多...

数值优化 (Numerical Optimization) 学习系列-线搜索方法 (LineSearch) (http://blog.c...

数值优化的学习过程是长期的、是枯燥的也是最有用的，一旦入门对机器学习者、算法工作者都会有很大的帮助。在此记录Numerical Optimization的学习、思考和实践。 ...

 fangqingan_java (http://blog.csdn.net/fangqingan_java) 2015年12月27日 18:44  4234



浅谈对梯度下降的理解 (http://blog.csdn.net/zhulf0804/article/details/52250220)

浅谈梯度下降法 如果读者对方向导数和梯度的定义不太了解，请先阅读上篇文章《方向导数与梯度》。 前些时间接触了机器学习，发现梯度下降法是机器学习里比较基础又比较重要的一个求最小值的算法。梯度下降...

 zhulf0804 (http://blog.csdn.net/zhulf0804) 2016年08月19日 14:04  4432



[机器学习] ML重要概念：梯度 (Gradient) 与梯度下降法 (Gradient Descent) (http://bl...

本文介绍机器学习中重要的概念：梯度和梯度下降法，这是我们在学习MachineLearning算法时的核心概念之一，其实也就是我们在大学本科高等数学中的基础概念。 ...

 waliik (http://blog.csdn.net/waliik) 2016年03月25日 13:34  11579



Mysql 笔记 (二) (<http://blog.csdn.net/u012339706/article/details/77930220>)

Mysql 笔记 (二) 本文仅是作为学习过程中记录笔记 (Mysql菜鸟教程学习的整理) ...

 u012339706 (<http://blog.csdn.net/u012339706>) 2017年09月11日 10:12  52

深度学习概述 (<http://blog.csdn.net/u012339706/article/details/77946994>)

深度网络的概述：包括其相对于浅层网络的优势，其训练的难处，及训练方法等的简单概率...

 u012339706 (<http://blog.csdn.net/u012339706>) 2017年09月12日 16:47  78



bp神经网络实例：贝叶斯、梯度下降算法 (<http://download.csdn.net/do...>)

(<http://download.csdn.net/do...>) 2010年07月09日 16:21 8KB [下载](#)




梯度下降算法 (http://download.csdn.net/download/lhd_paul/101166...)

(http://download.csdn.net/download/lhd_paul/101166...) 2017年11月13日 16:19 377KB [下载](#)

一种使用随机抽样梯度下降算法来预估词汇量的方法 (<http://blog.csdn.net/dotedy/article/...>)

我们经常可以看到各种各样的英语词汇量测试功能，你测试过吗？你觉得准吗？我使用过有道词典的词汇量测试功能，我认为它最大的问题是，不管是谁不管测多少次，每次测的词都是固定不变的，这就好像高考，全国...

 dotedy (<http://blog.csdn.net/dotedy>) 2015年12月22日 01:14  631



机器学习-数据挖掘-梯度下降算法C++实现 (<http://download.csdn.net/d...>)

(<http://download.csdn.net/d...>) 2011年06月16日 17:26 1.11MB [下载](#)





梯度下降算法 (<http://download.csdn.net/download/expensun/70815...>)

(<http://download.csdn.net/download/expensun/70815...>) 2014年03月22日 14:56 814B [下载](#)

机器学习_线性回归，梯度下降算法与正规方程 (<http://blog.csdn.net/matrix5267/article/...>)

个人对这方面的理解，文字纯手打，图片来自于coursera的课件 1.线性回归的定义：给出若干的训练集(训练集中 $x^{(j)}$ 表示样本中第j个项)，然后拟合为一条直线，使得cost最...

 matrix5267 (<http://blog.csdn.net/matrix5267>) 2016年12月20日 21:08  412



梯度下降算法动态演示matlab文件 (<http://download.csdn.net/downloa...>)

(<http://download.csdn.net/downloa...>) 2013年04月26日 15:51 1021B [下载](#)





机器学习小组知识点4：批量梯度下降算法(BGD) (<http://download.csdn.n...>)

(<http://download.csdn.n...>) 2016年10月18日 17:15 521KB [下载](#)

[action] deep learning 深度学习 tensorflow 实战(2) 实现简单神经网络以及随机梯度下降算..

在之前的实战(1) 中，我们将数据清洗整理后，得到了'notMNIST.pickle'数据。 本文将阐述利用tensorflow创建一个简单的神经网络以及随机梯度下降算法。 # These are ...

 u013805817 (<http://blog.csdn.net/u013805817>) 2016年08月04日 17:39  5721



梯度下降算法（有纤细的中文解释）(<http://download.csdn.net/download...>)

[/http://download...](http://download.csdn.net/download...)

2009年07月09日 15:13 29KB [下载](#)