

交叉熵代价函数（作用及公式推导）

2016年04月02日 18:22:52 阅读数：65167

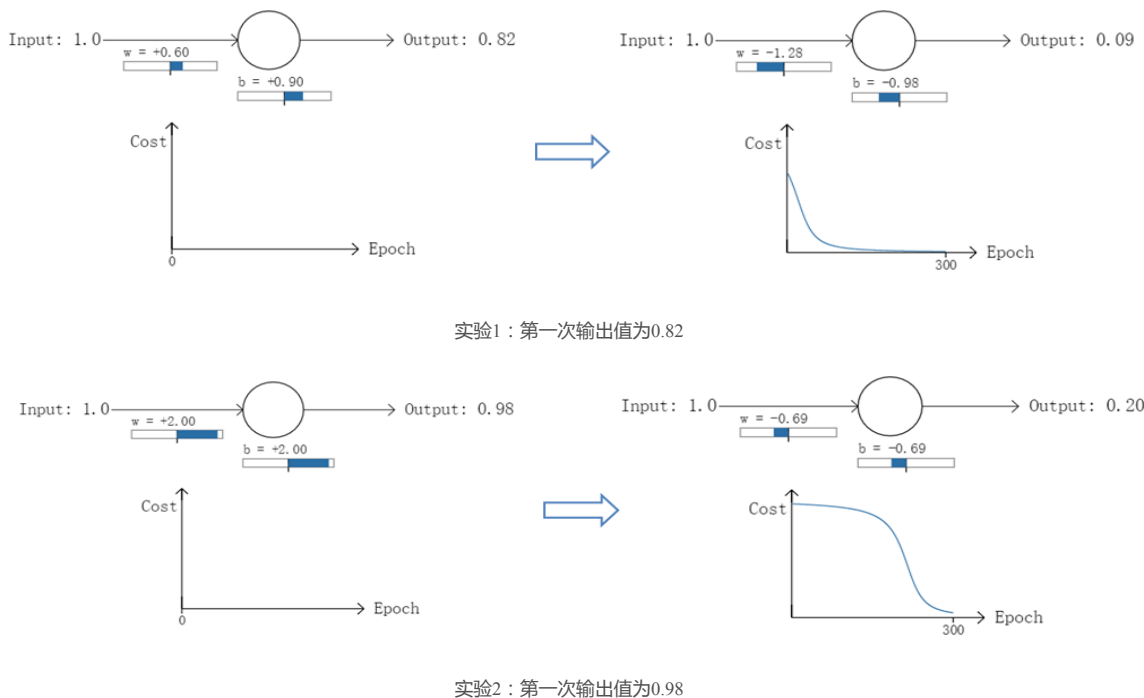
版权声明：本文为博主原创文章，未经博主允许不得转载。 <https://blog.csdn.net/u014313009/article/details/51043064>

交叉熵代价函数（Cross-entropy cost function）是用来衡量人工神经网络（ANN）的预测值与实际值的一种，与二次代价函数更有效地促进ANN的训练。在介绍交叉熵代价函数之前，本文先简要介绍二次代价函数，以及其存在的不足。

1. 二次代价函数的不足

ANN的设计目的之一是为了使机器可以像人一样学习知识。人在学习分析新事物时，当发现自己犯的错误，改正的力度就投篮：当运动员发现自己的投篮方向离正确方向越远，那么他调整的投篮角度就应该越大，篮球就更容易投进。同理，我们希望训练时，如果预测值与实际值的误差越大，那么在反向传播训练的过程中，各种参数调整的幅度就要更大，从而使训练更快收敛。然而二次代价函数训练ANN，看到的实际效果是，如果误差越大，参数调整的幅度可能更小，训练更缓慢。

以一个神经元的二类分类训练为例，进行两次实验（ANN常用的激活函数为sigmoid函数，该实验也采用该函数）：输入一个相同 $x=1.0$ （该样本对应的实际分类 $y=0$ ）；两次实验各自随机初始化参数，从而在各自的第一次前向传播后得到不同的输出值，形成不同的差）：



在实验1中，随机初始化参数，使得第一次输出值为0.82（该样本对应的实际值为0）；经过300次迭代训练后，输出值由0.82降到实际值。而在实验2中，第一次输出值为0.98，同样经过300迭代训练，输出值只降到了0.20。

从两次实验的代价曲线中可以看出：实验1的代价随着训练次数增加而快速降低，但实验2的代价在一开始下降得非常缓慢；直观的误差越大，收敛得越缓慢。

其实，误差大导致训练缓慢的原因在于使用了二次代价函数。二次代价函数的公式如下：

$$C = \frac{1}{2n} \sum \|y(x) - a^L(x)\|^2$$

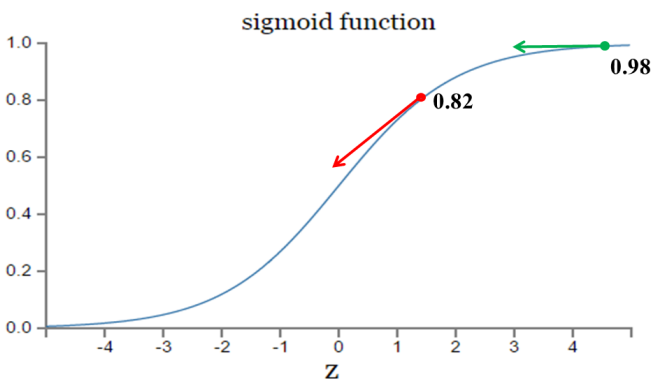
其中，C表示代价，x表示样本，y表示实际值，a表示输出值，n表示样本的总数。为简单起见，假设只有一个样本，此时n=1。那么，代价函数为：

$$C = \frac{(y - a)^2}{2}$$

目前训练ANN最有效的算法是反向传播算法。简而言之，训练ANN就是通过反向传播代价，以减少代价为目的，来调整参数。参数包括神经元之间的连接权重w，以及每个神经元本身的偏置b。调参的方式是采用梯度下降算法（Gradient descent），其核心思想是沿着梯度方向调整参数。b的梯度推导如下：

$$\frac{\partial C}{\partial w} = (a - y)\sigma'(z)x$$
$$\frac{\partial C}{\partial b} = (a - y)\sigma'(z)$$

其中，z表示神经元的输入，σ表示激活函数。从以上公式可以看出，w和b的梯度跟激活函数的梯度成正比，激活函数的梯度越大，参数调整得越快，训练收敛得就越快。而神经网络常用的激活函数为sigmoid函数，该函数的曲线如下所示：



如图所示，实验2的初始输出值（0.98）对应的梯度明显小于实验1的输出值（0.82），因此实验2的参数梯度下降得比实验1慢。这导致实验2的代价（误差）越大，导致训练越慢的原因。与我们的期望不符，即：不能像人一样，错误越大，改正的幅度越大，从而学习得越快。

可能有人会说，那就选择一个梯度不变化或变化不明显的激活函数不就解决问题了吗？图样图森破，那样虽然简单粗暴地解决了这个问题，但可能会引起其他更多更麻烦的问题。而且，类似sigmoid这样的函数（比如tanh函数）有很多优点，非常适合用来做激活函数，具体请参看相关资料。

2. 交叉熵代价函数

换个思路，我们不换激活函数，而是换掉二次代价函数，改用交叉熵代价函数：

$$C = -\frac{1}{n} \sum_x [y \ln a + (1 - y) \ln(1 - a)]$$

其中，x表示样本，n表示样本的总数。那么，重新计算参数w的梯度：

$$\begin{aligned}\frac{\partial C}{\partial w_j} &= -\frac{1}{n} \sum_x \left(\frac{y}{\sigma(z)} - \frac{(1-y)}{1-\sigma(z)} \right) \frac{\partial \sigma}{\partial w_j} \\ &= -\frac{1}{n} \sum_x \left(\frac{y}{\sigma(z)} - \frac{(1-y)}{1-\sigma(z)} \right) \sigma'(z) x_j \\ &= \frac{1}{n} \sum_x \frac{\sigma'(z) x_j}{\sigma(z)(1-\sigma(z))} (\sigma(z) - y) \\ &= \frac{1}{n} \sum_x x_j (\sigma(z) - y)\end{aligned}$$

其中（具体证明见附录）：

$$\sigma'(z) = \sigma(z)(1 - \sigma(z))$$

因此，w的梯度公式中原来的 $\sigma'(z)$ 被消掉了；另外，该梯度公式中的 $\sigma(z) - y$ 表示输出值与实际值之间的误差。所以，当误差就越大，参数w调整得越快，训练速度也就越快。同理可得，b的梯度为：

$$\frac{\partial C}{\partial b} = \frac{1}{n} \sum_x (\sigma(z) - y)$$

实际情况证明，交叉熵代价函数带来的训练效果往往比二次代价函数要好。

3. 交叉熵代价函数是如何产生的？

以偏置b的梯度计算为例，推导出交叉熵代价函数：

$$\begin{aligned}\frac{\partial C}{\partial b} &= \frac{\partial C}{\partial a} \cdot \frac{\partial a}{\partial z} \cdot \frac{\partial z}{\partial b} \\ &= \frac{\partial C}{\partial a} \cdot \sigma'(z) \cdot \frac{\partial (wx + b)}{\partial b} \\ &= \frac{\partial C}{\partial a} \cdot \sigma'(z) \\ &= \frac{\partial C}{\partial a} \cdot a(1 - a)\end{aligned}$$

在第1小节中，由二次代价函数推导出来的b的梯度公式为：

$$\frac{\partial C}{\partial b} = (a - y) \sigma'(z)$$

为了消掉该公式中的 $\sigma'(z)$ ，我们想找到一个代价函数使得：

$$\frac{\partial C}{\partial b} = (a - y)$$

即：

$$\frac{\partial C}{\partial a} \cdot a(1-a) = (a-y)$$

对两侧求积分，可得：

$$C = -[y \ln a + (1-y) \ln(1-a)] + \text{constant}$$

而这就是前面介绍的交叉熵代价函数。

附录：

sigmoid函数为：

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

可证：

$$\begin{aligned} \sigma'(z) &= \left(\frac{1}{1 + e^{-z}}\right)' \\ &= (-1)(1 + e^{-z})^{(-1)-1} \cdot (e^{-z})' \\ &= \frac{1}{(1 + e^{-z})^2} \cdot (e^{-z}) \\ &= \frac{1}{1 + e^{-z}} \cdot \frac{e^{-z}}{1 + e^{-z}} \\ &= \frac{1}{1 + e^{-z}} \cdot \left(1 - \frac{1}{1 + e^{-z}}\right) \\ &= \sigma(z)(1 - \sigma(z)) \end{aligned}$$

老股民酒后无意说漏：20年炒股 坚持只看1指标

第六 · 爍燚

想对作者说点什么？

我来说两句

 粼粼小淇


写的很好，请问交叉熵函数就是这么得到吗？（1周前 #15楼）

 ymmccc

太棒了，写的非常清楚，棒棒的（1周前 #14楼）

https://blog.csdn.net/u014313009/article/details/51043064?utm_source=blogxgwz0

4/9

 **cicido**：两种方式其实都包含了误差，二次函数中多了一项sigmoid的导数，在二函数损失推导过程a-y就是误差。所以关键在sigmoid的导数。只要证明其最大值少于1
前 #13楼)

查看 23 条热评

交叉熵为何能作损失函数

在很多二分类问题中，特别是正负样本不均衡的分类问题中，常使用交叉熵作为loss对模型的参数求梯度进行更新...

简单易懂的softmax交叉熵损失函数求导

来写一个softmax求导的推导过程，不仅可以给自己理清思路，还可以造福大众，岂不美哉~ softmax经常被添加在...



一年几十块钱的虚拟主机,你敢用吗

百度广告

一文搞懂交叉熵在机器学习中的使用，透彻理解交叉熵背后的直觉

关于交叉熵在loss函数中使用的理解 交叉熵 (cross entropy) 是深度学习中常用的一个概念，一般用来求目标与预...

机器学习基础（六）—— 交叉熵代价函数（cross-entropy error）

交叉熵代价函数比较困惑的一点是，Logistic Regression 中存在交叉熵作为误差函数，神经网络 (neural network) ...

交叉熵（Cross-Entropy）

交叉熵（Cross-Entropy）交叉熵是一个在ML领域经常会被提到的名词。在这篇文章里将对这个概念进行详细的分...

交叉熵代价函数(损失函数)及其求导推导

前言交叉熵损失函数交叉熵损失函数的求导 前言

白杨股王8年追涨停铁律“1272”曝光，震惊众人

第六· 熾燚

交叉熵

文章转自：http://blog.csdn.net/rtygbwwerr/article/details/50778098 交叉熵（Cross-Entropy）交叉熵是一个在ML...

关于交叉熵

在信息论中可以根据香农熵估计一个字符串中每个符号 S_{-i} 编码所需的最小位数，前提是每个字符出现的概率 $P...$

如何直观理解交叉熵及其优势？

在统计学中，损失函数是一种衡量系统错误程度的函数。而在有监督学习模型里，损失函数则是衡量模型对样本预...

交叉熵代价函数

1.从方差代价函数说起 代价函数经常用方差代价函数（即采用均方误差MSE），比如对于一个神经元（单输入单输...

相关热词 交叉熵最大似然 交叉熵大白话 均方误差和交叉熵 kl交叉熵 交叉熵均值

博主推荐



oio328Loio
[关注](#) 37篇文章



老司机的诗和远方
[关注](#) 542篇文章



那些很冒险的梦
[关注](#) 24篇文章

Backpropagation算法（反向传播算法）+cross-entropy cost（交叉熵代价函数）

Softmax函数与交叉熵

Softmax函数与交叉熵

深度学习（一）cross-entropy和sofrmax

Cross-entropy 神经网络的学习行为和人脑差的很多，开始学习很慢，后来逐渐增快 为什么？学习慢 => 偏导数 $\partial C/\partial w...$

中国海参大多数都是假的！10年女参农曝光30年行业丑闻！

三诚咨询· 熾燚

来自：wen 23 2万 的专栏

来自： 1.7万 的乐园

来自：史丹禾 2.2万 博客

来自：Zhang's Wikipedia 3.5万

来自：rtygbwwerr的专栏 8.5万

来自：liuheng0111的博客 1771

来自：菜鸟只能慢慢练 1.6万

来自：机器会学习的博客 1206

来自：对半独白 5957

来自：xueli1991的博客 232

来自：qq_30232405的博客 450

来自：behamcheung的博客 2.7万

来自：静默、 1.2万

深度学习基础理论探索(二)：cross-entropy损失函数的前世今生

5254

前面我们讲到，克服梯度消失有两个方向，1.用rule激活函数。2.改进损失函数用cross_entropy。 ok,首先我们先看...

来自： qq_25552539的博客

代价函数

一，什么是代价函数 我在网上找了很长时间代价函数的定义，但是准确定义并没有，我理解的代价函数就是用于找...

来自： justT 23 的博文

简单谈谈Cross Entropy Loss

写在前面 分类问题和回归问题是监督学习的两大种类。 神经网络模型的效果及优化的目标是通过损失函数来定义的...

来自： 杂货店 2835 的博文

通俗理解交叉熵

作者：CyberRep链接：https://www.zhihu.com/question/41252833/answer/195901726来源：知乎著作权归作者所有...

来自： AC 123 的博文

交叉熵以及为什么用交叉熵作为代价函数

作者：知乎用户链接：https://www.zhihu.com/question/41252833/answer/108777563来源：知乎著作权归作者所有...

来自： inne 1075 的博文



停车场管理系统

百度广告

交叉熵损失函数

最近看到CNNH(CNN Hashing)，里面有提到交叉熵损失函数，找了一下，以备查忘 转载 http://blog.csdn.net/u012...

来自： Losteng的博客 7747

神经网络学习（八）优化方法：交叉熵代价函数

回顾 上一节完成了BP神经网络的Python实现，并稍微改进了分类效果。下面是三层网络 [784,30,50,10]，mini_batc...

来自： ch的博文 161

下载 BP反向传播神经网络介绍及公式推导

07-08

神经网络中的反向传播（BackPropagation）介绍及公式推导 神经网络中的激活函数的作用，最常用的两个激活函数Sigmoid 和TanH 代价函数对介绍二次代价函数(Quadratic Cost)...

交叉熵的解决代价函数为二次函数导致学习慢问题（S型激活函数）

1189

1.从方差代价函数说起 代价函数经常用方差代价函数（即采用均方误差MSE），比如对于一个神经元（单输入单输...

来自： Touch_Dream的博客

深度学习框架TensorFlow学习与应用（三）——使用交叉熵作为代价函数

944

在TensorFlow中用：tf.nn.sigmoid_cross_entropy_with_logits()来表示跟sigmoid搭配使用的交叉熵。 tf.nn.softmax...

来自： JluTiger316的博客

2018高铁纪念币1:1等值兑换火爆开售中，仅限最后3天！！

如艺·顶新

01-逻辑回归的代价函数(交叉熵代价函数)以及梯度下降法

573

1.1 逻辑回归的代价函数 1.2 逻辑回归中的梯度下降法 参考文献 吴恩达老师Deep Learning课程...

来自： 琪琪的博客

简单的交叉熵损失函数，你真的懂了吗？

8191

个人网站：红色石头的机器学习之路 CSDN博客：红色石头的专栏 知乎：红色石头 微博：RedstoneWill的微博 Git...

来自： 红色石头的专栏

交叉熵、相对熵及KL散度通俗理解

1455

原文转载自http://blog.csdn.net/u012162613/article/details/44239919本文是《Neural networks and deep learning》...

来自： aoanng的博客

【机器学习】代价函数，损失函数，目标函数区别

4032

一：损失函数，代价函数，目标函数定义首先给出结论：损失函数（Loss Function）是定义在单个样本上的，算的...

来自： ly1771857509的博客

BP神经网络——从二次代价函数（Quadratic cost）到交叉熵（cross-entropy cost）代价函数

6737

我们首先来回顾BP神经网络反向传播过程：1. 首先在当前网络状态（(w,b)(w,b)给定）下，根据feedforward(a=σ(w...

来自： Zhang's Wikipedia



一年几十块钱的虚拟主机,你敢用吗

百度广告

反向传播算法推导

2094

BP(backpropgationalgorithm)：后向传导算法，顾名思义就是从神经网络的输出(顶层)到输入(底层)进行求解。那么...

来自： li8zi8fa的博客

结合斯坦福深度学习教程 对于BP反向传播算法原理的推导过程解析

3281

最近在对卷积神经网络（CNN）进行学习的过程中，发现自己之前对反向传播算法的理解不够透彻，所以今天专门...

模型评估与选择（后篇）-代价曲线

2202

看博客上都在说书上关于代价曲线的介绍太过于简单，只考书中的内容无法理解，我当时看完竟然咻有这种感觉，...

来自： qingyun_wudaoletu的博客

深度学习系列-交叉熵代价函数

346

交叉熵代价函数基本完成了些许期末考试，特意来补一些softmax，因为涉及到损失函数。损失函数一般分为mse，...

来自： Cutedipin的博客

五、改进神经网络的学习方法（1）：交叉熵代价函数

1011

本文主要介绍了改进神经网络的学习方式中的第一种方法，采用交叉熵函数代替均方误差函数作为代价函数。首先...

来自： dugunzhang的博客

十月起，白杨上班族可申读成人大学，1年学完，国家认可

升学教育 · 顶新

解释为什么BP神经网络中有人使用二次代价函数，而有的人使用交叉熵

89

这段时间一直在看有关于神经网络的教程，在之前看过的其他一些机器学习算法，例如logistics回归等都是用C=(y...

来自： weixin_40512345的博客

softmax交叉熵作为损失函数在神经网络中的作用与求导

80

softmax 函数softmax(柔性最大值)函数，一般在神经网络中， softmax可以作为分类任务的输出层。其实可以认为s...

来自： 的博客

简述为什么输出层在使用sigmoid做激励函数时, 代价函数用交叉熵比平方差更好

219

符号： C：代价函数 w：weight b：bias η ：学习速率 在使用平方差作代价函数时：

来自： leung的博客

多分类问题中的交叉熵

4492

熵的本质是香农信息量()的期望。现有关于样本集的2个概率分布p和q，其中p为真实分布，q非真实分布。按照真...

来自： 往者不可谏，来者犹可追

交叉熵是什么，能干什么用，

246

在做tensorflow对手写mnist字体识别中，用到了交叉熵来定义损失函数的方法，看着别人用这个，但是又不知道原...

来自： weixin_38859557的博客



多线线切割机的切线是怎么回事阿

百度广告

softmax的log似然代价函数（公式求导）

3.8万

在人工神经网络（ANN）中，softmax通常被用作输出层的激活函数。这不仅是因为它的效果好，而且因为它使得A...

来自： Arthur-Chen的专栏

多目标及多分类区别及交叉熵问题

2338

多目标即事件是独立的，但不要求互斥。可以认为事件发生在多个样本空间，彼此互不影响。例问一张图片中包括...

来自： m0_37753130的博客

TensorFlow四种Cross Entropy算法实现和应用

5659

作者：陈迪豪，就职于小米，负责企业深度学习平台搭建，参与过HBase、Docker、OpenStack等开源项目，目前...

来自： CSDN 人工智能

深度学习中常用的代价函数

1901

1.二次代价函数(quadratic cost)：其中，C表示代价函数，x表示样本，y表示实际值，a表示输出值，n表示样本的...

来自： chen645096127的博客

交叉熵代价函数（cross-entropy cost function）

1.1万

1.从方差代价函数说起 代价函数经常用方差代价函数（即采用均方误差MSE），比如对于一个神经元（单输入单输...

来自： wtq1993的博客

早上还只喝白开水吗？加一物排出全身毒素，医生也建议经常喝！

河北森和矿 · 熨熨

神经网络与深度学习 笔记4 交叉熵代价函数 softmax函数

417

1. 学习缓慢问题 二次代价函数,定义如下 其中 a 是神经元的输出,训练输入为 x = 1,y = 0 则是目标输出。显式地使用...

来自： qq_35497808的博客

机器学习-梯度下降III（为何梯度下降能最小化代价函数）

485

我们已经知道梯度下降算法的定义，就是不断的进行下面的操作，假设 θ_0, θ_1 已被初始化为0 直到J(θ_0, θ_1)收敛...

来自： sword_csdn的博客

【机器学习入门】Andrew NG《Machine Learning》课程笔记之二：基本概念、代价函数、梯度下降和线...

953

机器学习基本概念在开始正式的内容学习之前，先了解一些关于机器学习的基本概念《机器学习》的作者Tom Mitch...

来自： luckybuzhi的博客

TensorFlow学习（四）代价函数、拟合、优化器

74

二次代价函数 相当于把误差平方累加再除以样本总数，求平均

来自： Gloria971111的博客

代价函数和激活函数组合效果

均方差代价函数 + sigmoid $C = \frac{1}{2} \sum (y - a)^2$ 其中, $z = wx + b$, 而 a 是经历了激活函数的 z , $a = \sigma(z)$ 来自: [念去去的博客](#)



云服务器 试用

百度广告


有关熵的几个概念 及 最大似然和交叉熵的一致性

随机事件的信息量 为了描述一个随机事件的信息量，定义了自信息。自信息表示不确定性减少的程度。 来自: [yujianmi的专栏](#)

商务与经济统计（英文版·原书第13版）

本书是美国辛辛那提大学的安德森、斯威尼教授和罗切斯特理工学院的威廉斯教授再度合作的结晶。新版在保留了...

没有更多推荐了, [返回首页](#)



—鸿

关注

原创26

粉丝130

喜欢58


评论73

等级: 博客4


访问: 26万+

积分: 1424

排名: 4万+



停车场管理系统



最新文章

softmax的log似然代价函数（公式求导）

反向传播算法（过程及公式推导）

模式识别与机器学习（二）：常用的概率分布（共轭分布等）

模式识别与机器学习（一）：概率论、决策论、信息论

红黑树的插入与删除

个人分类

机器学习7篇

观点挖掘1篇

主题建模2篇

Hadoop7篇

推荐算法2篇

展开

归档

2016年4月3篇

2016年3月1篇

2016年2月	1篇
2015年8月	1篇
2014年10月	4篇
展开	



联系我们



扫码联系客服



下载CSDN APP

 kefu@csdn.net

 400-660-0108

 QQ客服

 客服论坛

关于我们 招聘 广告服务 网站地图

 百度提供站内搜索 京ICP证09002463号

©2018 CSDN版权所有

经营性网站备案信息 网络110报警服务

北京互联网违法和不良信息举报中心

中国互联网举报中心

