

## hello小工的专栏

≡ 目录视图

≡ 摘要视图

RSS 订阅

个人资料



hello小工

访问: 47486次

积分: 723

等级: BLOG &gt; 3

排名: 千里之外

原创: 8篇 转载: 120篇

译文: 0篇 评论: 7条

## BP神经网络梯度下降算法

2016-06-23 18:21

3111人阅读

评论(4)

收藏

举报

目录(?)

[+]

菜鸟初学人智相关问题，智商低，艰苦学习中，转文只为保存，其中加上了一些个人注释，便于更简单的理解~新手也可以看，共勉。

转自博客园@编程De：http://www.cnblogs.com/jzhlin/archive/2012/07/28/bp.html

## 从神经网络的生物模型说起

我们知道人脑信息的传递、对外界刺激产生反应都由神经元控制的，人脑就是由上百亿个的这样神经元构成。这些神经元之间并不孤立而且联系很密切，每个神经元平均与几千个神经元相连接，因此构成了人脑的神经网络。刺激在神经网络中的传播是遵循一定的规则的，一个神经元并非每次接到其他神经传递过来的刺激都产生反应。它首先会将与其相邻的神经元传来的刺激进行积累，到一定的时候产生自己的刺激将其传递给一些与它相邻的神经元。这样工作的百亿个的神经元构成了人脑对外界进行的反应。而人脑对外界刺激的学习的机制就是通过调节这些神经元之间联系以及其强度。当然，实际上以上说的是对人脑真正神经工作的一种简化的生物模型，利用这种简化的生物模型可以将它推广至机器学习中来，并把它描述成人工神经网络。BP神经网络就是其中的一种，来看看具体对神经元的分析。

文章搜索

文章存档

[2016年10月](#) (1)[2016年08月](#) (3)[2016年07月](#) (2)[2016年06月](#) (9)[2016年05月](#) (14)

展开

阅读排行

[RNN介绍，较易懂](#) (7230)[BP神经网络梯度下降算法](#) (3087)[10个深度学习的工具](#) (2921)[haartraining训练生成xm](#) (1633)[使用scikit-learn进行机器](#) (1410)[海量数据采集爬虫架构](#) (1316)[踩内存](#) (1202)[数据清洗实例分析](#) (1079)[Wellner 自适应阈值二值](#) (1053)[端到端的OCR：基于CNI](#) (866)

评论排行

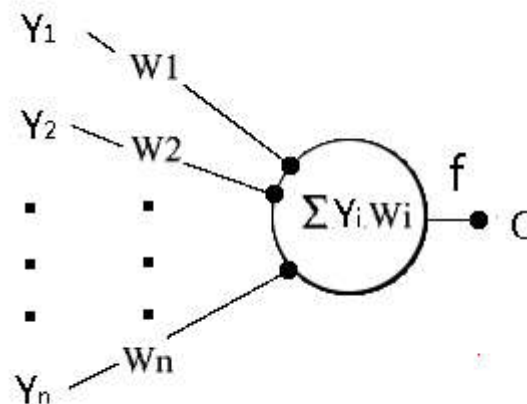
[BP神经网络梯度下降算法](#) (4)

图1 神经网络中神经元示意图

神经元的积累的刺激是由其他神经元传递过来的刺激量和对应的权重之和，用  $X_j$  表示这种积累， $Y_i$  表示某个神经元传递过来的刺激量， $W_i$  表示链接某个神经元刺激的权重，得到公式：

$$X_j = (Y_1 * W_1) + (Y_2 * W_2) + \dots + (Y_i * W_i) + \dots + (Y_n * W_n)$$

而当  $X_j$  完成积累后，完成积累的神经元本身对周围的一些神经元传播刺激，将其表示为  $y_j$  得到如下所示：

$$y_j = f(X_j)$$

神经元根据积累后  $X_j$  的结果进行处理后，对外传递刺激  $y_j$ 。用  $f$  函数映射来表示这种处理，将它称之为 **激活函数**。

## BP神经网络的构成

分析完单个的神经元后，再来看看它们组成网络后的情形，用图形来说明是最直观的方法，如图2所示：

- RNN介绍, 较易懂 (2)
- 15个java多线程面试题 (1)
- C++面试题 (0)
- C++内存管理 (0)
- c++面试题 (0)
- C语言堆栈入门——堆和 (0)
- perl expect (0)
- java面试题 (0)
- C语言内存地址分析 (0)

#### 推荐文章

- \* 探索通用可编程数据平面
- \* 这是一份很有诚意的 **Protocol Buffer** 语法详解
- \* **CSDN日报20170420** ——《开发和产品之间的恩怨从何处来?》
- \* **Android**图片加载框架最全解析——从源码的角度理解**Glide**的执行流程
- \* 如果两个程序员差不多, 选写作能力更好的那个
- \* 从构造函数看线程安全

#### 最新评论

- BP神经网络梯度下降算法**  
**1875151847**: 没有错啦, 这里是求解解释的很好, 非常清楚 thanks
- RNN介绍, 较易懂**  
暖暖的阳光ncepu: 赞一个
- RNN介绍, 较易懂**

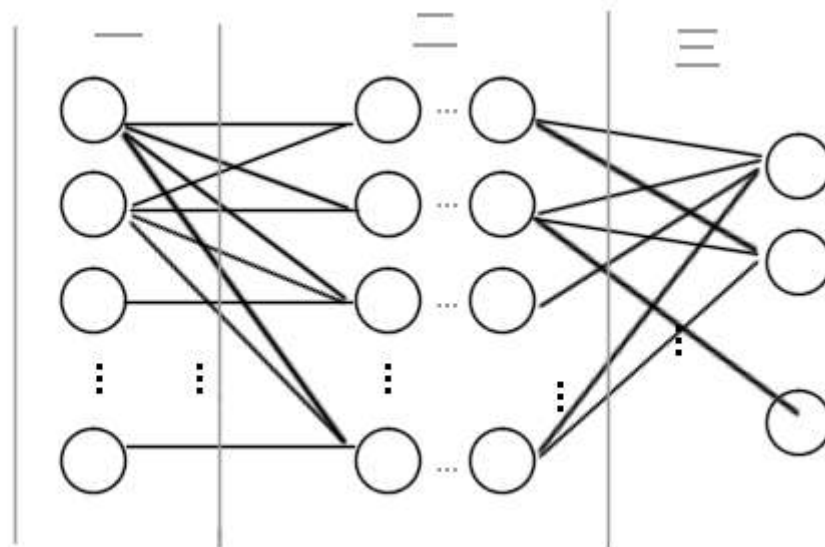


图2 BP神经网络示意图

第一区域的来说, 它们相当于外界的刺激, 是刺激的来源并且将刺激传递给神经元, 因此把第一区域命名为输入层。第二区域, 表示神经元相互之间传递刺激相当于人脑里面, 因此把第二区命名为隐藏层。第三区域, 表示神经元经过多层次相互传递后对外界的反应, 因此把第三区域命名为输出层。

简单的描述就是, 输入层将刺激传递给隐藏层, 隐藏层通过神经元之间联系的强度(权重)和传递规则(激活函数)将刺激传到输出层, 输出层整理隐藏层处理的后的刺激产生最终结果。若有正确的结果, 那么将正确的结果和产生比较, 得到误差, 再逆推对神经网络中的链接权重进行反馈修正, 从而来完成学习的过程。这就是BP神经网的反馈机制。也正是BP(Back Propagation)名字的来源: 运用向后反馈的学习机制, 来修正神经网络中的权重, 最终达到输出正确的!

## BP神经网络的数学推导

从数学上对BP神经网络模型进行分析, 本文第一部分神经网的生物模型中可以得到关于BP神经网络的第一个公式(1):

qq\_25892947: 你这搬运有点不合格啊，原文是《理解 LSTM 网络》，你的标题是RNN介绍，较易懂

BP神经网络梯度下降算法

我师父微微: 楼主 爱你，搜bp只有你把这个导数怎么推导的写出来了

BP神经网络梯度下降算法

我师父微微: 错了 E是y和d的差的平方

BP神经网络梯度下降算法

我师父微微: 错了 E是y和d的差的平方

15个java多线程面试题

helloworldsss: 棒棒哒

$$x_j = \sum_i y_i w_{ji}$$

对于神经元本身的输出的激活函数，一般来说选取 Sigmoid 函数，那么可以得到第二个公式（2）：

$$y_j = \frac{1}{1 + e^{-x_j}}$$

弱逼补充：为啥选择这个激活函数呢？因为这个函数连续可导，平滑性质好，

具体看图：



至于为啥不选择其他同样平滑的函数，who knows...目测是实践中检验出来的。好下面接着看原作者的

通过以上两个公式，可以分析出来BP神经网络中输出结果的计算过程。每个神经元收到刺激  $y_i$  然后加权重  $w_{ji}$  完成后产生  $x_j$ ，再通过激活函数产生刺激  $y_j$ ，向下一层与它相连的神经元传递，依次类推最终输出结果。

我们再来分析如何利用向后反馈机制来修正神经元权重  $w_{ji}$ ，这一部分数学推导需要运用到多元微分的数学内容。要修正  $w_{ji}$  就需要得到误差量。具体来看，首先用  $d_j$  来表示真实的正确结果，并且设误差为  $E$ ，那么  $(y_j - d_j)$  对应的就是  $E$  对于  $y_j$  的微分增量，即  $y_j$  减去  $(y_j - d_j)$  后就能得到正确值，得到公式（3）：

$$\frac{\partial E}{\partial y_j} = y_j - d_j$$

接着补充：这里提到了个微分增量，我把数学忘完了。。。

建议百度百科，这里微分就相当于E这个错误是和y<sub>i</sub>的值成函数关系，y<sub>i</sub>-d<sub>i</sub>就是函数关系，y<sub>i</sub>是自变量，E是函数的结果，所以这么一个表示就是微分。。。后面的公示也是同理嘞。

同时，下面的证明过程中，一定要注意下标呀。j表示的是当前层。i表示的是前一层，记住这个后面就容易理解。

$$\frac{\partial E}{\partial w_{ji}}$$

然后，明确目标，需要知道的是对于权重 W<sub>ji</sub> 的误差量是多少也就是  $\frac{\partial E}{\partial w_{ji}}$  的值。而由公式（1）中知道 W<sub>ji</sub> 与 x<sub>j</sub> 相关，那么可以推导出公式（4）：

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial x_j} \frac{\partial x_j}{\partial w_{ji}} = \frac{\partial E}{\partial x_j} y_i$$

$$\frac{\partial E}{\partial x_j}$$

需要求得 W<sub>ji</sub> 的误差量，转换为需要求  $\frac{\partial E}{\partial x_j}$  的值了。它的推导如下：

$$\frac{\partial E}{\partial x_j} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial x_j}$$

$$\frac{\partial y_j}{\partial x_j}$$

其中  $\frac{\partial y_j}{\partial x_j}$  的值，可以通过公式（2）求导得出来：

$$\frac{\partial y_j}{\partial x_j} = y_j(1 - y_j)$$

所以最终得到的的误差量的值为：

$$\frac{\partial E_j}{\partial w_{ji}} = y_j(1 - y_j)(y_j - d_j)y_i$$

以上公式需要注意下标：最后一个为  $y_i$ ，前面的都是  $y_j$ 。推到这里可以算是完成了运用神经网络的输出值  $y_j$  和正确值  $d_j$  对最后一层隐藏层  $w_{ji}$  的修正，那么对其他隐藏层呢？接着往下看。

$$\frac{\partial E_j}{\partial y_i}$$

上面的推导过程由公式（3）开始，如果我们知道（注意是  $y_i$ ，公式（3）中是  $y_j$ ），就可以同理推导出其对应其他隐藏层需要修正的权重值误差量了。推导如下：

$$\frac{\partial E_j}{\partial y_i} = \frac{\partial E_j}{\partial y_j} \frac{\partial y_j}{\partial x_j} \frac{\partial x_j}{\partial y_i} = (y_j - d_j)y_j(1 - y_j)w_{ji}$$

这样所有的误差量的都可以同理推导完成！

这步推到只推出了三个连乘的第一个，后面两个值与那层的变量相关，随意结果只是下标改改。

最后一步修正  $w_{ji}$ ，就是加上下面变量了，设置一个  $\eta$ （0 到 1 之间）学习率。

$$\Delta W = -l \frac{\partial E}{\partial w}$$

为啥要乘上呢呢，理解是一点点的去减少调整这个误差，设置值太小，慢，设置值太大，步长就大，这 $l$ 也应该是一个值得考虑的问题。

至此，BP神经网络反馈部分的数学推导算完成了，可以在自己的草稿纸上画画~

写在后面：梯度下降有个问题是容易陷入局部最小值的问题。如果这个问题不了解的话，那可以回想回想一个3d平面，有若干坑，如果随机点沿着下降最快的地方去跑，很可能跑到了一个小坑里面，却没有跑到真正的全局最小值的坑里。

从博文的评论中看到该问作者实现了这个神经网络，但是没有使用这个梯度下降的模型。呵呵，这个就算是理解什么是神经网络，什么是从后面往前推导的过程。下面待学习的就是评论中提到的动量因子模型。

作者实现实现了动量因子模型，代码：

[http://www.cnblogs.com/jzhlin/archive/2012/07/30/bp\\_c.html](http://www.cnblogs.com/jzhlin/archive/2012/07/30/bp_c.html)

[http://www.cnblogs.com/jzhlin/archive/2012/08/01/bp\\_c2.html](http://www.cnblogs.com/jzhlin/archive/2012/08/01/bp_c2.html)

以及还要看的：

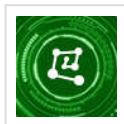
概率神经网络 (PNN) <http://www.cnblogs.com/jzhlin/archive/2012/07/28/bp.html>

顶 踩  
0 0

上一篇 10个深度学习的工具

下一篇 各种算法的优点与缺点

参考知识库



机器学习知识库

17699 关注 | 2162 收录



人工智能机器学习知识库

477 关注 | 297 收录

## 猜你在找

[阿里云机器学习算法应用实践](#)[BP神经网络的简单字符识别算法](#)[人工智能之机器学习算法的介绍](#)[WEKA算法解析 -- 多层感知器BP神经网络](#)[Python数据分析实战：泰坦尼克号之灾与机器学习算法](#)[文本分类算法之一--BP神经网络](#)[使用 K-近邻算法识别简单数字验证码实战](#)[从神经网络到BP算法纯理论推导](#)[统计机器学习入门——线性模型选择与正则化1](#)[斯坦福机器学习-week5 学习笔记1神经网络训练BP算法](#)[查看评论](#)4楼 [l875151847](#) 2017-04-11 11:45发表

没有错啦，这里是求导  
解释的很好，非常清楚  
thanks

3楼 [我师父微微](#) 2017-03-14 17:21发表

楼主 爱你，搜bp只有你把这个导数怎么推导的写出来了

2楼 [我师父微微](#) 2017-03-14 17:14发表**[cpp]**

01. 错了 E是y和d的差的平方

1楼 [我师父微微](#) 2017-03-14 17:14发表





错了 E是y和d的差的平方

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

#### 核心技术类目

全部主题   Hadoop   AWS   移动游戏   Java   Android   iOS   Swift   智能硬件   Docker   OpenStack  
VPN   Spark   ERP   IE10   Eclipse   CRM   JavaScript   数据库   Ubuntu   NFC   WAP   jQuery  
BI   HTML5   Spring   Apache   .NET   API   HTML   SDK   IIS   Fedora   XML   LBS   Unity  
Splashtop   UML   components   Windows Mobile   Rails   QEMU   KDE   Cassandra   CloudStack  
FTC   coremail   OPhone   CouchBase   云计算   iOS6   Rackspace   Web App   SpringSide   Maemo  
Compuware   大数据   aptech   Perl   Tornado   Ruby   Hibernate   ThinkPHP   HBase   Pure   Solr  
Angular   Cloud Foundry   Redis   Scala   Django   Bootstrap

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)

网站客服   杂志客服   微博客服   webmaster@csdn.net   400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 |

江苏乐知网络技术有限公司

京 ICP 证 09002463 号 | Copyright © 1999-2016, CSDN.NET, All Rights Reserved



2