

阿凡卢

博客园 首页 新随笔 联系 订阅 管理

随笔- 138 文章- 0 评论- 381

BP人工神经网络的介绍与实现

神经网络概念与适合领域

神经网络最早的研究是 40 年代心理学家 Mcculloch 和数学家 Pitts 合作提出的，他们提出的MP模型拉开了神经网络研究的序幕。

神经网络的发展大致经过 3 个阶段：1947~1969 年为初期，在这期间科学家们提出了许多神经元模型和学习规则，如 MP 模型、HEBB 学习规则和感知器等；60 年代末期至 80 年代中期，神经网络控制与整个神经网络研究一样，处于低潮。在此期间，科学家们做了大量的工作，如 Hopfield 教授对网络引入能量函数的概念，给出了网络的稳定性判据，提出了用于联想记忆和优化计算的途径。1984年，Hiton 教授提出 Bol tzman 机模型；1986年 Kumelhart 等人提出误差反向传播神经网络，简称 BP 网络。目前，BP网络已成为广泛使用的网络。1987年至今为发展期，在此期间，神经网络受到国际重视，各个国家都展开研究，形成神经网络发展的另一个高潮。

人工神经网络（ANN）受到生物学的启发是生物神经网络的一种模拟和近似，它从结构、实现机理和功能上模拟生物神经网络。从系统观点看，人工神经网络是由大量神经元通过极其丰富和完善的连接而构成的自适应非线性动态系统。人工神经网络，因为生物的学习系统是由相互连接的神经元组成的异常复杂的网络，其中每一个神经元单元有一定数量的实值输入，并产生单一的实数值输出。1960 年威德罗和霍夫率先把神经网络用于自动控制研究。神经网络以其独特的结构和处理信息的方法，在许多实际应用领域中取得了显著的成效，主要应用如下：自动控制领域、处理组合优化问题、模式识别、图像处理、传感器信号处理、机器人控制、信号处理、卫生保健、医疗、经济、化工领域、焊接领域、地理领域、数据挖掘、电力系统、交通、军事、矿业、农业和气象等领域。

神经网络基本结构

人工神经网络由神经元模型构成，这种由许多神经元组成的信息处理网络具有并行分布结构。每个神经元具有单一输出，并且能够与其它神经元连接；存在许多（多重）输出连接方法，每种连接方法对应一个连接权系数。可把 ANN 看成是以处理单元 PE(processing element) 为节点，用加权有向弧(链)相互连接而成的有向图。令来自其它处理单元(神经元)i的信息为Xi，它们与本处理单元的相互作用强度为 Wi，i=0,1,..., n-1，处理单元的内部阈值为 θ。那么本神经元的输入为：

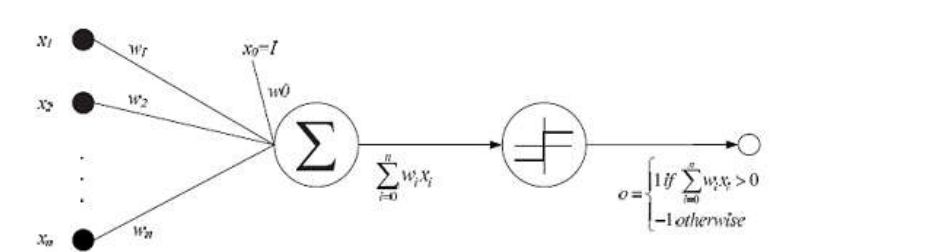
$$\sum_{i=0}^{n-1} w_i x_i$$

而处理单元的输出为：

$$y = f(\sum_{i=0}^{n-1} w_i x_i - \theta)$$

式中，xi为第 i 个元素的输入，wi 为第 i 个元素与本处理单元的互联权重。f 称为激发函数(activation function)或作用函数。它决定节点(神经元)的输出。该输出为 1 或 0 取决于其输入之和大于或小于内部阈值 θ。

下图所示神经元单元由多个输入Xi，i=1,2,...,n和一个输出y组成。中间状态由输入信号的权和表示，而输出为：



训练网络

神经网络结构被设计完成，有了输入、输出参数后，我们就要对网络进行训练。神经网络的训练有包括感知器训练、delta 规则训练和反向传播算法等训练，其中感知器训练是基础。

感知器和 delta 训练规则

理解神经网络的第一步是从对抽象生物神经开始,本文用到的人工神经网络系统是以被称为感知器的单元为基础，如图所示。感知器以一个实数值向量作为输入，计算这些输入的线性组合，如果结果大于某个阈值，就输出 1，否则输出 -1，如果 x 从 1 到 n，则感知器计算公式如下：

公告

昵称：阿凡卢  
园龄：3年6个月  
粉丝：349  
关注：35  
+加关注

2012年12月						
日	一	二	三	四	五	六
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

搜索

常用链接

我的随笔  
我的评论  
我的参与  
最新评论  
我的标签  
更多链接

随笔分类(139)

- Algorithm(25)
- Big Data(6)
- C#(11)
- C/C++(19)
- Data Structure(15)
- DataBase(2)
- Distributed System(2)
- GIS(13)
- Java(15)
- Linux(3)
- Network(7)
- Program Design(4)
- Python(2)
- Research(6)
- Web(2)
- Windows(7)

随笔档案(138)

- 2015年12月 (2)
- 2015年10月 (3)
- 2015年8月 (2)
- 2015年7月 (3)
- 2015年5月 (1)

$$o(x_1, \dots, x_n) = \begin{cases} 1 & w_0 + w_1x_1 + \dots + w_nx_n > 0 \\ -1 & otherwise \end{cases}$$

其中每个 **wi** 是一个实数常量，或叫做权值，用来决定输入 **xi** 对感知器输出的贡献率。特别地，**-w0**是阈值。

尽管当训练样例线性可分时，感知器法则可以成功地找到一个权向量，但如果样例不是线性可分时它将不能收敛，因此人们设计了另一个训练法则来克服这个不足，这个训练规则叫做 **delta** 规则。感知器训练规则是基于这样一种思路--权系数的调整是由目标和输出的差分方程表达式决定。而 **delta** 规则是基于梯度降落这样一种思路。这个复杂的数学概念可以举个简单的例子来表示。从给定的几点来看，向南的那条路径比向东那条更陡些。向东就像从悬崖上掉下来，但是向南就是沿着一个略微倾斜的斜坡下来，向西象登一座陡峭的山，而北边则到了平地，只要慢慢的闲逛就可以了。所以您要寻找的是到达平地的所有路径中将陡峭的总和减少到最小的路径。在权系数的调整中，神经网络将会找到一种将误差减少到最小的权系数的分配方式。这部分我们不做详细介绍，如有需要大家可参考相关的人工智能书籍。

反向传播算法

人工神经网络学习为学习实数值和向量值函数提供了一种实际的方法，对于连续的和离散的属性都可以使用。并且对训练数据中的噪声具有很好的健壮性。反向传播算法是最常见的网络学习算法。这是我们所知用来训练神经网络很普遍的方法，反向传播算法是一种具有很强学习能力的系统，结构比较简单，且易于编程。

鲁梅尔哈特(Rumelhart)和麦克莱兰(Meclelland)于 1985 年发展了 BP 网络学习算法，实现了明斯基的多层网络设想。BP网络不仅含有输入节点和输出节点，而且含有一层或多层隐(层)节点。输入信号先向前传递到隐藏节点，经过作用后，再把隐藏节点的输出信息传递到输出节点，最后给出输出结果。节点的激发函数一般选用 **S** 型函数。

反向传播(back-propagation, BP)算法是一种计算单个权值变化引起网络性能变化值的较为简单的方法。由于BP算法过程包含从输出节点开始，反向地向第一隐含层(即最接近输入层的隐含层)传播由总误差引起的权值修正，所以称为"反向传播"。反向传播特性与所求解问题的性质和所作细节选择有极为密切的关系。

对于由一系列确定的单元互连形成的多层网络，反向传播算法可用来学习这个多层网络的权值。它采用梯度下降方法试图最小化网络输出值和目标值之间的误差平方，因为我们要考虑多个输出单元的网络，而不是像以前只考虑单个单元，所以我们要重新计算误差**E**，以便对所有网络输出的误差求和：

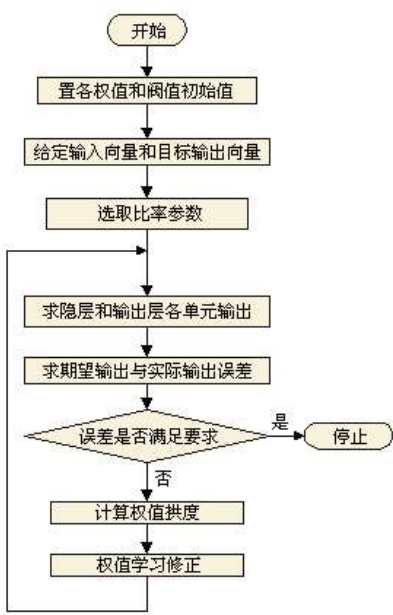
$$E(\vec{w}) = \frac{1}{2} \sum_{d \in D} \sum_{k \in \text{outputs}} (t_{kd} - o_{kd})^2$$

**Output** 是网络输出单元的集合，**tkd** 和 **okd** 是与训练样例 **d** 和第 **k** 个输出单元的相关输出值。

反向传播算法的一个迷人特性是：它能够在网络内部的隐藏层发现有用的中间表示：

- 1.训练样例仅包含网络输入和输出，权值调节的过程可以自由地设置权值，来定义任何隐藏单元表示，这些隐藏单元表示在使误差**E**达到最小时最有效。
- 2.引导反向传播算法定义新的隐藏层特征，这些特征在输入中没有明确表示出来，但能捕捉输入实例中与学习目标函数最相关的特征

反向传播训练神经元的算法如下：



C++简单实现与测试

以下C++代码实现了BP网络，通过8个3位二进制样本对应一个期望输出，训练BP网络，最后训练好的

- 2015年4月 (2)
- 2015年3月 (1)
- 2015年2月 (1)
- 2015年1月 (1)
- 2014年12月 (2)
- 2014年11月 (1)
- 2014年10月 (3)
- 2014年9月 (3)
- 2014年8月 (1)
- 2014年7月 (3)
- 2014年6月 (3)
- 2014年5月 (2)
- 2014年4月 (1)
- 2014年3月 (3)
- 2014年1月 (1)
- 2013年12月 (6)
- 2013年11月 (2)
- 2013年10月 (4)
- 2013年9月 (4)
- 2013年8月 (2)
- 2013年7月 (2)
- 2013年6月 (1)
- 2013年5月 (4)
- 2013年4月 (3)
- 2012年12月 (4)
- 2012年11月 (10)
- 2012年10月 (14)
- 2012年9月 (15)
- 2012年8月 (28)

友情链接

- IBM developerworks
- UC技术博客
- 阿里中间件团队博客
- 酷壳
- 美团技术团队博客
- 奇虎360技术博客
- 淘宝核心系统团队博客
- 淘宝搜索技术博客
- 腾讯ISUX

最新评论

- 1. Re:RabbitMQ的几种典型...  
~感谢分享~  
--jerry\_0824
- 2. Re:Mybatis关联查询（嵌...  
N+1 查询的那个N可不可以是在缓存中查询  
--zonghua
- 3. Re:百度谷歌离线地图解...  
希望博主本着共享精神，能把此源码发布出来 共享一下。  
--huangb\_99
- 4. Re:基于Libevent的HTTP...  
@zzmarquis建议用第2种方式吧...  
--阿凡卢
- 5. Re:基于Libevent的HTTP...  
请问，想多线程http server。想你说的两种方式：1、启多个线程，每个子线程持有一个event\_base和evhttp，接受请求和处理业务逻辑在同一个子线程中。2、主线程持有一个event\_ba.....  
--zzmarquis

阅读排行榜

- 1. C++中数字与字符串之间...
- 2. C#多线程编程(32413)
- 3. NPOI读写Excel(27392)

网络可以将输入的三位二进制数对应输出一位十进制数。

 [View Code](#)

参考文献

神经网络介绍——利用反向传播算法的模式学习  
<http://www.ibm.com/developerworks/cn/linux/other/l-neural/index.html>

人工智能 Java 坦克机器人系列：神经网络，上部  
<http://www.ibm.com/developerworks/cn/java/j-lo-robocode3/index.html>

人工智能 Java 坦克机器人系列：神经网络，下部  
<http://www.ibm.com/developerworks/cn/java/j-lo-robocode4/>

使用 Python 构造神经网络--Hopfield 网络可以重构失真的图案并消除噪声  
<http://www.ibm.com/developerworks/cn/linux/l-neurnet/>

提供一个Matlab的BP神经网络的基础资料  
<http://www.cnblogs.com/galaxyprince/archive/2010/12/20/1911157.html>

codeproject上的一个实现，作者已经给出好几种形式的应用例子  
[http://www.codeproject.com/KB/recipes/aforge\\_neuro.aspx](http://www.codeproject.com/KB/recipes/aforge_neuro.aspx)

作者： 阿凡卢

出处：<http://www.cnblogs.com/luxiaoxun/>

本文版权归作者和博客园共有，欢迎转载，但未经作者同意必须保留此段声明，且在文章页面明显位置给出原文连接，否则保留追究法律责任的权利。

分类：**Research**

标签：BP , ANN

好文要顶

关注我

收藏该文







阿凡卢

关注 - 35

粉丝 - 349

+加关注

40

(请您对文章做出评价)

« 上一篇：[C++ STL 学习笔记](#)  
» 下一篇：[Windows下一个日志类](#)

posted @ 2012-12-10 14:25 阿凡卢 阅读(16440) 评论(7) 编辑 收藏

评论列表

- #1楼

2013-11-15 09:30 Sure::D

嗯，谢谢博主，以前总是在用matlab工具箱，其实能够写一写这样的代码还是挺好的！

支持(0) 反对(0)
- #2楼

2015-04-17 14:25 Fragrance307

博主，您好！看了您的代码，发现两个问题：  
1、权值更新应该是在求得所有层偏差之后，不应该是在求出输出层偏差后立刻更新隐含层--输出层的权值，然后再用该值去求输入层--隐含层的权值；  
2、测试样本不应该是训练样本集中的样本，而应该用训练集中没有过的。您代码中用的{1,1,1}是训练集中的样本，这样正确率肯定会高啊~~~

支持(1) 反对(0)
- #3楼

[楼主] 2015-04-17 19:09 阿凡卢 @Fragrance307

这仅仅是学习的代码，不是用来做实验的。

支持(0) 反对(0)

4. [Mybatis关联查询（嵌套...](#)
5. [Mybatis实现数据的增删...](#)
6. [基于Netty4的HttpServe...](#)
7. [C#读写config配置文件\(1...](#)
8. [BP人工神经网络的介绍与...](#)
9. [C++的构造函数和析构函...](#)
10. [C# byte数组与Image...](#)

评论排行榜

1. [百度谷歌离线地图解决方...](#)
2. [高斯混合模型GMM的C+...](#)
3. [GMap.Net开发之在地图...](#)
4. [GMap.Net开发之自定义...](#)
5. [NPOI读写Excel\(24\)](#)
6. [GMap.Net开发之在WinF...](#)
7. [基于GMap.Net的地图解...](#)
8. [基于netty-socketio的w...](#)
9. [GMap.Net开发之地址解...](#)
10. [基于VLC的播放器开发\(...](#)

推荐排行榜

1. [C#多线程编程\(14\)](#)
2. [NPOI读写Excel\(13\)](#)
3. [C# byte数组与Image的...](#)
4. [基于GMap.Net的地图解...](#)
5. [C++中数字与字符串之间...](#)
6. [C++中的new、operato...](#)
7. [百度谷歌离线地图解决方...](#)
8. [GMap.Net开发之在WinF...](#)
9. [几种服务器端IO模型的简...](#)
10. [基于Netty与RabbitMQ...](#)

#4楼

2015-04-17 19:12 Fragrance307

-----

嗯嗯，知道的，只是看了以后想到这两点，C++编程姚向博主学习的~~~

支持(0) 反对(0)

#5楼

2015-08-25 14:35 sjk1207

-----

博主 我问你哦 为什么计算理想值和现有输出值的偏差是这么算的？

qq[k]=(yd[k]-x2[k])\*x2[k]\*(1-x2[k]); //希望输出与实际输出的偏差

后面乘的二次函数，是认为接近边界时误差就适当缩小？（这个二次函数的极值在1/2嘛 所以我这么认为）

支持(0) 反对(0)

#6楼

[楼主] 2015-08-25 20:18 阿凡卢

-----

@sjk1207

这是读研时候的文章，细节已记不清，请参考相关论文。

支持(0) 反对(0)

#7楼

2016-01-08 13:33 gystld

-----

推荐一款最易用中文excel神经网络软件包，有BP、RBF、PNN、GRNN、Elman、SOM、LVQ七种神经网络功能。一键操作，易用易懂。<http://www.plsexcelword.com/>。

支持(0) 反对(0)

刷新评论 刷新页面 返回顶部

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

- 【推荐】50万行VC++源码：大型组态工控、电力仿真CAD与GIS源码库
- 【推荐】融云即时通讯云—专注为 App 开发者提供IM云服务
- 【阿里云SSD云盘】速度行业领先



- 最新IT新闻：
- 忘了键盘吧！谷歌已开发出语音录入文档工具
  - 清华硕士IT男虚充700万加油卡获刑14年半
  - 五维黑洞“黑环”可能存在：或颠覆广义相对论
  - Canonical创始人：考虑让Ubuntu支持鸚鵡桌面
  - ZergHelper：成功绕过iOS App Store的代码审查的风险软件
- » 更多新闻...

- 最新知识库文章：
- 谷歌背后的数学
  - Medium开发团队谈架构设计
  - 理解“渐进增强(Progressive Enhancement)”
  - 为什么说DOM操作很慢
  - 为什么你应该尝试全栈
- » 更多知识库文章...