

# 多层前向神经网络的权值平衡算法

裴浩东, 苏宏业, 褚 健

(浙江大学工业控制技术国家重点实验室, 浙江杭州 310027)

摘 要: 对前向多层神经网络BP算法的改进方法作了分析,提出了一种新的改进方法——权值平衡算法,充分发挥各权值对网络训练的作用.仿真结果表明,新算法与基本的BP算法相比明显地提高了网络收敛速度和精度,对适合运用本算法的神经网络结构作了探讨.

关键词: BP算法;神经网络;权值平衡算法

中图分类号: TP13 文献标识码: A 文章编号: 0372-2112 (2002) 01-0139-03

## Weight Balance Algorithm of Multilayer Neural Network

PEI Hao-dong, SU Hong-ye, CHU Jian

(National Lab. of Industrial Process Control, Zhejiang University, Hangzhou, Zhejiang 310027, China)

Abstract: The improved BP algorithm of multilayer neural network is discussed and a new method, weight balance algorithm is proposed to exert all weights effect on the process of neural network training. Simulations indicate that it has a superior convergence rate and precision compared to the standard BP algorithm. The neural network structure which fits for this algorithm is also discussed.

Key words: BP algorithm; neural network; weight balance algorithm

### 1 引言

多层前向神经网络作为一种非线性系统的辨识工具得到了广泛的应用,目前用于多层前向神经网络计算的算法很多,但仍以BP算法为主.因BP算法本质上是一种梯度法,如果搜索步长选择不恰当,收敛速度将会很慢,且易于陷入局部极小点.为此,许多研究者提出了改进的算法,主要是围绕四个方面进行改进<sup>[1-5]</sup>:(1)改进学习率参数调节方法,如学习率大小随误差梯度而变化;(2)改变激励函数,如把Sigmoid函数修正成分段函数;(3)权值修正方法,如动量项法、牛顿法;(4)改变误差函数,如采用柯西误差估计器的  $E_{总} = \sum_{p=1}^M \sum_{k=1}^K \ln[1 + (t_k^p - o_k^p)^2]$  等.各种改进算法对提高BP算法收敛和克服局部最小均有进步意义,但也有一定的局限性,有的改进方法效果不很明显,有的方法非常复杂,然而复杂的方法又给算法收敛和计算维数带来新的问题,不易实现,调节参数也较困难.本文从应用角度出发,提出一种简单有效的方法,能大大地提高神经网络的收敛速度和精度.

### 2 基于BP算法的权值平衡法

#### 2.1 BP算法分析

多层神经网络的结构如图1所示(以三层网络为例),输入矢量为  $x_0, x_1, \dots, x_{n-1}$ ;第二层有  $n_1$  个神经元,输出层有  $m$  个神经元,输入层与第二层之间权值为  $w_{ik}$ ,第二层与输出层之间的权为  $w_{kl}$ ,为方便起见,把阈值写入连接权中去,则:

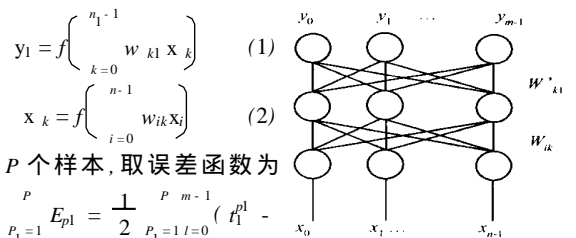


图1 多层前向神经网络

设有  $P$  个样本,取误差函数为  $E_{总} = \sum_{p=1}^P E_{pl} = \frac{1}{2} \sum_{p=1}^P \sum_{l=0}^{m-1} (t_l^{pl} - y_l^{pl})^2$  采用梯度法,可得到每一层的权值迭代公式,输出层与第二层权值:

$$w_{kl}(n_0 + 1) = w_{kl}(n_0) + \sum_{p=1}^P \frac{P_1}{kl} X_k \frac{P_1}{k}; \quad (3)$$
$$\frac{P_1}{kl} = (t_l^p - y_l^p) y_l^p (1 - y_l^p)$$

输入层与第二层权值:

$$w_{ik}(n_0 + 1) = w_{ik}(n_0) + \sum_{p=1}^P \frac{P_1}{ik} X_k \frac{P_1}{i}; \quad (4)$$
$$\frac{P_1}{ik} = \left( \sum_{k=0}^m \frac{P_1}{kl} w_{kl} \right) X_k (1 - X_k)$$

整个网络学习过程分为两个阶段<sup>[6]</sup>,第一阶段是网络前传运算,这一部分可实现对系统的非线性映射能力.第二阶段是对权值和阈值的修改算法,由上面的分析可知,网络前传时,先计算第一层单元的输出,再计算第二层的输出,最后计算最外层单元的输出.权值修正时,先调整输出层到第二层的

收稿日期:2000-08-21;修回日期:2001-05-28

基金项目:国家杰出青年科学基金(No. 60025308)

权值和阈值,再调整第二层到输入层的权值和阈值,故称误差反传(BP)算法。

在BP算法的训练过程中,观察一下一个典型的3层BP网络的权值,我们发现输出层与隐层之间的权值调节量要远大于隐层与输入层的权值调节量,表1是一个3-5-1结构的三层神经网络在学习过程中的权值调节量。

表1 网络层间权值调节量比较

输入层与隐层	0.002600	0.001800	-0.00296	-0.00168	-0.00211
间权值调节量	0.002500	0.001747	-0.00287	-0.00163	-0.00205
$w_{ik}$	0.002631	0.001821	-0.00299	-0.00169	-0.00213
隐层与输出层					
间权值调节量	-0.01653	-0.02145	-0.03181	-0.03201	-0.02711
$w_{kl}$					

可见,输入层与隐层值之间的权值调节量  $w_{ik}$  和隐层与输出层之间权值调节量  $w_{kl}$  的绝对值大小有明显差别,相差之大超过一个数量级。这是因为在式(4)中,初始化时权值  $w_{kl}$  比较小,而  $x_k^{p_1} \in [0, 1]$ ,  $x_k^{p_1}(1-x_k^{p_1}) \leq 0.25$ , 所以  $\frac{p_1}{ik} < \frac{p_1}{kl}$ , 即隐层与输入层间权值调节量小于隐层与输出层间权值调节量,随着权值的不断变化,输入层与隐层之间的权值调节量会有所增大,而一般权值达不到很大,另一方面随着学习次数的增加,  $(t_i^{p_1} - y_i^{p_1}) y_i^{p_1} (1 - y_i^{p_1})$  变小,  $\frac{p_1}{kl}$  也变得较小。从而使  $\frac{p_1}{ik}$  一直处于较小状态,使输入层和隐层之间权值对网络训练的贡献很小。图2显示了各层权值调节量在神经网络训练过程中的变化规律。虚线是隐层与输出层之间权值调节量随学习次数的变化,实线是输入层与隐层之间权值调节量。

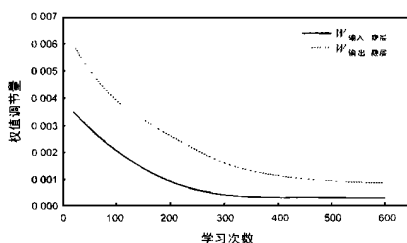


图2 权值调节量的变化曲线

在神经网络训练过程中,权值调节量太大易产生振荡,太小则收敛速度太慢。由以上分析可见,输入层与隐层值之间的权值调节量和隐层与输出层之间权值调节量对网络训练的贡献有明显差别,相差之大超过一个数量级。当隐层与输出层之间权值调节量处于合适值时,输入层与隐层值之间的权值调节量则太小,不能有效地参与学习过程;当输入层与隐层值之间的权值调节量处于合适值时,隐层与输出层之间权值调节量太大,易产生过调。因此,为加快网络收敛速度,应使二者在同一个数量级内,使他们对网络训练有同等的贡献,协调一致地加快网络训练的速度。

## 2.2 权值平衡法

以3层神经网络为例,在隐层激励函数中引入  $k$  因子:

$$x_k^{p_1} = \frac{1}{1 + \exp(-)} = \frac{1}{1 + \exp[- (w_{ik} x_i^{p_1}) / k]}$$

$$p_{ik}^{p_1} = \left( \prod_{l=0}^{m-1} p_{kl}^{p_1} w_{kl} \right) x_k^{p_1} (1 - x_k^{p_1}) / k \quad (5)$$

该式与式(4)相比,多出了一个  $1/k$  因子影响 sigmoid 函数的形态,当  $k < 1$  时,激励函数的梯度增加,加快网络收敛速度,当  $k > 1$  时,可使梯度变小,函数曲线变得平坦。 $k$  因子还能平衡输出层与中间层权值调节量和输入层与中间层权值调节量的巨大差别,取  $k < 1$ , 输入层与中间层权值调节量是输出层与中间层权值调节量的  $1/k$  倍,两层权值调节量不平衡状况得到解决。同时注意到  $k$  引入使激励函数变陡,易陷入平坦区,这里采用网络前传和反向误差计算分开来处理,既避免了陷入平坦区的问题,又解决了权值的平衡问题。算法总结如下:

- (1) 取  $\frac{1}{k} = \frac{(1 - w_{kl})}{(1 - w_{ik})}$ ;
- (2) 取适当的学习率,使  $w_{kl}$  处于合适值;
- (3) 在神经网络前传运算时,  $k$  保持不变,以保证网络对系统的正确辨识;
- (4) 在误差反传运算时,根据步骤(1)动态变化  $k$ 。

## 3 仿真结果

把这种改进的算法与传统BP、动量项法、自适应斜率法进行比较,分别用于解决异或问题、正弦函数逼近、焊接过程建模,都有很好的效果。下面数据为10次的平均值。

表2 收敛速度比较

	解决异或问题	正弦函数逼近	焊接过程建模
均方误差	0.0002	0.02	0.005
标准BP法	25次	680次	1120次
动量项法	23次	600次	1000次
自适应斜率法	22次	610次	1090次
权值平衡法	22次	490次	750次

由上表可见,权值平衡法比其它方法收敛速度提高很多,在用各种方法辨识模型时,能达到的最佳逼近程度也不相同。通过对焊接过程建模,作了大量实验,表3给出了部分样本和几种改进方法在3-5-1结构的网络上学习收敛后所达到的值的比较。

表3 收敛精度比较

样本	标准BP	动量项	自适应斜率	权值平衡
0.869	0.8502	0.8601	0.8533	0.8673
0.657	0.6888	0.6672	0.6772	0.6588
0.354	0.3850	0.3620	0.3763	0.3526
0.253	0.2519	0.2521	0.2431	0.2497
0.331	0.3633	0.3413	0.3526	0.3472
0.366	0.3765	0.3690	0.3601	0.3630
0.401	0.3720	0.3831	0.3921	0.3883
0.461	0.4356	0.4425	0.4398	0.4619
0.453	0.4275	0.4401	0.4337	0.4390
0.463	0.4487	0.4567	0.4512	0.4692

可见,这种改进方法与其他几种方法相比,有更高的收敛精度。

## 4 网络结构选择的影响

神经网络的收敛是依靠权值的不断变化而完成的对非线性

性系统的映射,对于各权值参与网络训练的贡献情况引入一个权值贡献率的概念,定义网络权值贡献率为调节权在所有连接权值中所占的比重.设三层前向神经网络神经元个数依次为  $n_1-n_2-n_3$ ,则所有的权值个数为  $n_1 \times n_2 + n_2 \times n_3$ ,神经网络权值贡献率  $M = \frac{(n_2 \times n_2) + k(n_1 \times n_2)}{n_1 \times n_2 + n_2 \times n_3}$ ;  $k$  为与最外层调节权相比权值调节量系数,通常  $k < 1$ ,所以  $M < 1$ ,权值平衡算法就是设法使  $M = 1$ .当  $k < 0.1$  时,网络训练主要依靠  $n_2 \times n_3$  个权值的贡献.由此可见,当输出端  $n_3 = 1$  时,这种结构的网络权值贡献率最小,  $n_1$  越大被弱化的权值越多,网络权值贡献率也越小,这种情况采用本算法更为有效.当神经网络为四层时,有两个隐层,可以在隐层的激励函数中分别引入  $k$  和  $j$ ,来平衡各层之间权值调节量.四层网络 ( $n_1-n_2-n_3-n_4$ )

权值贡献率为  $\frac{n_3 \times n_4 + k_2(n_2 \times n_3) + k_1(n_1 \times n_2)}{n_1 \times n_2 + n_2 \times n_3 + n_3 \times n_4}$ ,  $k_1 < k_2 < 1$ ,与三层网络相比四层网络中权值贡献率更小,所以权值平衡算法在四层网络中也更为有效.对焊接过程用四层神经网络(3-8-8-1)进行了训练,结果表明:当均方误差达到 0.005 时,标准 BP 算法用了 1322 次,而本算法只用了 702 次.比较表二可看出,四层神经网络中本算法的优越性更为显著.

权值平衡法优点归纳为:(1)方法简单,易于实现.(2)能显著提高网络的收敛速度和精度.(3)易于和其他改进方法结合使用.(4)神经网络层数越多,优势越明显,更适合于多输入神经元和单输出神经网络的学习.

#### 参考文献:

- [ 1 ] Engozinger S, Tomsen E. An accelerated learning algorithm for multilayer perceptions: Optimization layer by layer [J]. IEEE Trans on Neural Networks, 1995, 6(1): 31 - 42.

- [ 2 ] Ghellinck G de, Vial J P. A polynomial newton method for linear programming [J]. Algorithmica, 1986, 1(3): 425 - 453.
- [ 3 ] Karayiannis N J, Venetsanopoulos A N. Fast learning algorithm for neural networks [J]. IEEE Trans. Cas-1, 1992, 39(7): 453 - 474.
- [ 4 ] Robert S Scalero, Nazif Tepedelenlioglu. A fast new algorithm for training feedforward neural networks [J]. IEEE Transactions of Signal Processing, 1992, 40(1): 202 - 210.
- [ 5 ] 刘光远, 邱玉辉. 基于稳健误差估计器的快速 BP 算法 [J]. 计算机科学, 1997, 24(2): 66 - 68.
- [ 6 ] 张立明. 人工神经网络的模型及其应用 [M]. 上海: 复旦大学出版社, 1993.

#### 作者简介:



**裴浩东** 男. 1970 年 7 月生于安徽省铜陵市. 1993 年毕业于合肥工业大学, 1996 年甘肃工业大学硕士毕业后到浙江大学任教, 1998 年在浙江大学攻读博士学位. 主要从事计算机控制、人工智能、优化等方面的工作. E-mail: hdpei@ipc.zjn.edu.cn



**苏宏业** 男. 1969 年生. 1990 年毕业于南京化工大学, 1993 年获浙江大学工业自动化专业硕士学位, 1995 年获浙江大学工业自动化专业博士学位. 现为浙江大学先进控制研究所副所长、教授. 主要研究兴趣是鲁棒控制, 时滞系统控制, 非线性系统控制和过程控制理论和应用.