文章编号:1001 - 9081(2006)05 - 1106 - 03

双向权值调整的前向神经网络学习算法

陈华伟[1],年晓玲[2],靳　蕃[1]
(1. 西南交通大学 信息科学与技术学院,四川 成都 610031;
2. 中国平安保险股份有限公司 信息管理中心,上海 200040)
(chw. mail@ 163. com)

摘　要:提出一种新的前向神经网络的学习算法,该算法在正向和反向阶段对不同的层间权值进行必要的调整,在正向阶段按最小范数二乘解原则确定连接隐层与输出层的权值,反向阶段则按误差梯度下降原则调整连接输入层与隐层间的权值。新算法具有很快的收敛速度,并且能在一定的程度上保证所训练神经网络的泛化能力。实验结果初步验证了所提出算法的性能。

关键词:前向神经网络;双向权值调整;Moore-Penrose 广义逆
中图分类号: TP183　　**文献标识码**:A

Feedforward neural network learning algorithm based on bi-phases weights' adjusting

CHEN Hua-wei[1], NIAN Xiao-ling[2], JIN Fan[1]
(1. *School of Information Science & Technology, Southwest Jiaotong University, Sichuan Chengdu* 610031, *China;*
2. *Centre of Information Management, Ping An Insurance(Group) Company of China Ltd, Shanghai* 200040, *China*)

Abstract: A novel learning algorithm for feedforward neural networks was presented, which adjusts the weights during both forward phase and backward phase. In the forward pass, the least norm square solution is calculated as the values of the weights between the hidden layer and output layer, while the weights connecting the input layer to hidden layer are modified according to error gradient descent method during the back-propagation pass. The training algorithm adjusts different weights of the neural network during both passes, so it converged quickly and generalization ability can be improved due to the least norm square solution. The experimental results show that the new algorithm can produce good performance compared to Levenberg-Marquardt BP algorithm.

Key words: feedforward neural network; bi-phases weights' adjusting; Moore-Penrose generalized inverse

0　引言

前向神经网络是最常用的神经网络结构,在拓扑结构上表现为有向无环图。理论证明,仅含一个隐含层的三层前向 sigmoid 神经网络,只要隐节点数目足够多,就能以任意的精度逼近任意连续函数[1,2]。尽管通用逼近定理说明神经网络有一个隐层就能实现任意逼近,但实际上网络权值的合理配置是 NP 问题[3]。

误差反向传播算法(Error Back-propagation Algorithm, BP)是最常用的神经网络权值分配的监督学习算法。该算法基于最小均方误差原则,在反向阶段采用梯度下降法在权矢量空间中求取能够使误差函数极小化的权值作为学习问题的解答。虽然 BP 算法成功地解决前向多层神经网络的权值训练问题,但在实际应用中该算法存在着易陷入局部极小、学习收敛速度慢、易出现震荡现象等缺陷。因此,在 BP 的基础上提出了很多改进学习算法,典型的有共轭梯度算法[4]、Levenberg-Marquardt 算法[5](LMBP)等。其中 LMBP 既有高斯牛顿法的局部收敛性,又具有梯度下降法的全局特性,就训练速度及准确度而言,LMBP 明显优于其他算法。但所有 BP 算法及其改进算法都只是在反向阶段对网络的权值做出调整。

本文提出一种新的前向神经网络学习算法,该算法不但在反向阶段对输入权值进行必要的调整,而且在正向阶段对输出权值加以修正,具有较快的学习速度,并且能在一定的程度上保证神经网络的泛化能力。

1　广义逆与最小范数二乘解[6]

1.1　Moore-Penrose 广义逆

定义 1　设 $A = (a_{ij}) \in C^{m\times n}$,若 $n \times m$ 型矩阵 G 满足下列 Moore-Penrose 方程:

$$AGA = A,\ GAG = G,\ (GA)^H = GA,\ (AG)^H = AG \quad (1)$$

则称 G 为 A 的 Moore-Penrose 广义逆,记为 A^+。在所有的广义逆矩阵中,只有 A^+ 是存在且唯一。

1.2　最小范数二乘解

求解线性系统 $Ax = y$ 的解,A 可能奇异也可能是非方阵,但都可以通过使用 Moore-Penrose 广义逆来求解。

定义 2　对于线性系统 $Ax = y$,如果 $\hat{x}$ 满足

$$\|A\hat{x} - y\| = \min_x \|Ax - y\| \quad (2)$$

则 $\hat{x}$ 是最小二乘解。$\|\ \|$ 表示欧氏空间的范数。

定义 3　对于 $Ax = y$,若存在 $x_0 \in R^n$,满足:

$$\begin{cases} \|x_0\| \leqslant \|x\| \\ \forall x \in \{x: \|Ax - y\| \leqslant \|Az - y\|, \forall z \in R^n\} \end{cases}$$

则 x_0 称为线性系统 $Ax = y$ 的最小范数二乘解。

收稿日期:2005 - 11 - 14;**修订日期**:2006 - 02 - 23
作者简介:陈华伟(1972 -),男,江西宁都人,博士研究生,主要研究方向:计算智能、语音信号处理;　年晓玲(1978 -),女,内蒙古呼和浩特人,硕士,主要研究方向:计算机软件体系、网络安全;　靳蕃(1934 -),男,湖南长沙人,教授,博士生导师,主要研究方向:神经网络、信息编码.

定理1　对于线性系统$Ax = y$,如果存在一个矩阵G使得Gy是该系统的最小范数二乘解的充要条件是 $G = A^+$。即当$x = A^+ y$时既使得 $\| Ax - y \|$ 最小,同时x的范数亦最小。

2　单隐层前向神经网络的学习表示

2.1　单隐层前向神经网络

通用逼近定理说明前向神经网络有一个隐层就能实现任意逼近。单隐层前向神经网络的示意图如图1所示。为方便讨论,图示中将输出神经元层简化为仅含一个线性神经元,隐层神经元的激励函数为 *sigmoid* 函数。不失一般性,此模型可以推广到多输出非线性单调激励输出神经元的单隐层前向神经网络。

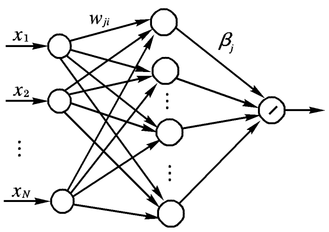

图1　单隐层前向神经网络结构

2.2　单隐层前向神经网络的学习问题

假设有N个不同的需要学习的样本对(x_i, t_i),

$x_i = [x_{i1}, x_{i2}, \cdots, x_{in}]^T$

$t_i = [t_{i1}, t_{i2}, \cdots, t_{im}]^T$

一个$\tilde{N}$个隐节点的单隐层神经网络,隐单元激励函数为$g(x)$,则对于这N个不同样本的学习问题可以表示为:

$$\sum_{i=1}^{\tilde{N}} \beta_i g(w_i \cdot x_j + b_i) = o_j, \qquad j = 1, 2, \cdots, N \tag{3}$$

$w_i = [w_{i1}, w_{i2}, \cdots, w_{in}]^T$为连接第$i$个隐节点与输入神经元的权矢量,$\beta_i = [\beta_{i1}, \beta_{i2}, \cdots, \beta_{im}]^T$为连接第$i$个隐单元与输出神经元的权矢量,$b_i$是第$i$个隐节点的阈值,$w_i \cdot x_j$表示$w_i$和$x_j$的内积。则当输入为$x_j$,第$i$个隐节点的输出为:

$$h_{ji} = g(w_i \cdot x_j + b_i) \tag{4}$$

现假设单隐层前向神经网络能够以零误差学习N个不同的样本,则(5)式成立:

$$\sum_{j=1}^{N} \| o_j - t_j \| = 0 \tag{5}$$

即存在下列矩阵形式:

$$H\beta = T \tag{6}$$

其中 $$H = \begin{bmatrix} h_{11} & \cdots & h_{1\tilde{N}} \\ \vdots & & \vdots \\ h_{N1} & \cdots & h_{N\tilde{N}} \end{bmatrix} \quad \beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{bmatrix} \quad T = \begin{bmatrix} T_1^T \\ \vdots \\ T_N^T \end{bmatrix}$$

H称为隐层输出矩阵。

2.3　输出权矢量的最小范数二乘解法

由(5)(6)可以知,对于确定输入权值和隐层阈值,神经网络可以建模为线性系统。此时训练前向神经网络的输出权值可以简化为求解$H\beta = T$的最小二乘解$\hat{\beta}$:

$$\| H\hat{\beta} - T \| = \min_{\beta} \| H\beta - T \| \tag{7}$$

基于最小二乘法思想,在径向基神经网络中采用了伪逆的方法用来确定输出层的权值,但伪逆并不是唯一的。

定理1显示满足(7)式的唯一的最小范数二乘解为$\hat{\beta} = H^+ T$,即Moore-Penrose广义逆是既能使神经网络输出与期望输出之间平方误差最小,同时为最小平方误差条件下范数最小的权矢量。Bartlett关于前向神经网络的泛化理论指出,对于获得小的学习误差的前向神经网络,权值的范数越小,则网络的泛化性能越好[7]。

3　双向权值调整学习算法

上节指出,对于给定的输入权值和隐层神经元的阈值,隐层输出矩阵确定,则$\hat{\beta} = H^+ T$是唯一的最小范数二乘解权矢量,即可以直接确定一个最优的隐层到输出层连接权值。基于Moore-Penrose广义逆和单隐层神经网络结构,Huang在文献[7]中提出Extreme Learning Machine的学习机制。分析发现,尽管$\hat{\beta}$是$H\beta = T$的最小范数二乘解,但对于任何给定的输入权值和隐层神经元的阈值,$\hat{\beta}$只是使得当前条件下的输出误差最小,并不能保证输出误差一定小于某一误差门限,或者说能使误差充分小。因此,结合BP算法的权值修正思想,本文提出正向确定输出权矢量(连接隐层和输出层的权矢量),反向调整输入权矢量(连接输入层和隐层的权矢量)的正、反两个阶段均能对网络权值加以调整的学习算法,称之为双向权值调整学习算法(Bi-Phases Weights' Adjusting, BPWA)。算法的原理示意图如图2所示。

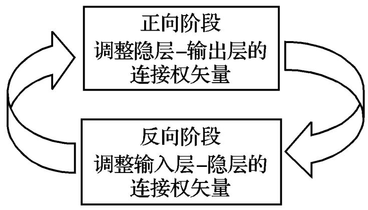

图2　双向权值调整的学习算法原理

从图2可以看出,双向权值调整的学习算法将权值的调整分成两个阶段,在正向和反向两个过程中分别调整网络不同部分的权矢量。

3.1　正向阶段基于Moore-Penrose广义逆的输出权值调整/确定算法

对于给定的输入层到隐层的权值和隐层单元阈值,根据所有需要学习的样本:

1) 计算隐层输出矩阵H;

2) 根据H,计算输出权值$\hat{\beta} = H^+ T$;

3) 计算$H\hat{\beta}$和T之间的误差,如果误差小于误差门限或者达到最大预定算法学习次数,算法停止;否则进入反向传播阶段。

3.2　反向阶段基于误差反向传播的输入权值调整算法

在反向阶段,使用误差梯度下降原则修改输入层到隐层的权值和隐层单元阈值。与传统BP不同的是隐层到输出层的权值$\hat{\beta}$在误差反向传播阶段不做修正,反向阶段仅调整输入层与隐层之间的连接权值及隐单元阈值。即输出层误差反向通过隐层到输出层之间的连接,传递到隐层单元,作为隐层单元分配的误差根据梯度下降原理修改输入层到隐层的连接权值和隐层单元阈值。

本文中输出层设为线性单元,隐层激励函数选用Sigmoid函数,则输入层与隐层间的连接权值(阈值作为输入为+1的权值)的调整公式可概括为:

$$\omega_{i\ j}^{k-1\ k}(t+1) = \omega_{i\ j}^{k-1\ k}(t) + \eta d_j^k O_i^{k-1} \tag{8}$$

综合以上分析,可以认为正向学习过程是一个快速的"粗调"过程,反向学习过程则是一个"细调"的过程。由于每次正向学习都会直接确定使输出误差最小化的最优输出权矢

量,当误差仍大于误差门限时,通过反向误差传递阶段,根据梯度下降调节输入层到隐层的权矢量和阈值,所以算法的收敛速度很快。

与以往算法不同在于,该算法通过正向确定最优输出权矢量和反向调整输入权矢量,可以使神经网络的训练即权值调整在正反两个阶段都实现。由于输出权矢量总是最小范数二乘解,根据 Bartlett 的理论[8]可知,连接隐层和输出层的权值在保证小的学习误差的情况下也尽量保证网络的泛化性能。

4 性能评价

通过函数逼近和分类问题实验,测试所提出的新算法,并将该学习算法与 LMBP 算法的实验性能加以比较。在以下的实验中,神经网络结构相同,仅采用不同的学习算法。学习误差和泛化性能以平均误差平方和作为性能指标,并以平均学习时间作为算法学习速度方面的衡量指标。所有实验在一台 CPU 为 P4 2.93GHz,内存容量为 1GB,操作系统为 Windows XP SP2 的机器上运行。在每次实验中,两种算法的误差目标值设为相同的期望学习误差值,最大迭代次数设置相同最大学习次数,每个测试的性能均为 100 次实验的平均性能。

4.1 Sinc 函数的逼近

Sinc 函数定义如下:

$$y(x) = \begin{cases} \sin(x)/x, & x \neq 0 \\ 1, & x = 0 \end{cases} \tag{9}$$

训练集和测试集分别为 5000 个表示为(x_i, y_i)的数据样本集合,x_i 是在[−10,10]区间上均匀分布随机产生。Sinc 函数训练集中的 y_i 均附加[−0.2,0.2]区间内均匀分布产生的噪声,测试集样本则为无噪声的 Sinc 数据。神经网络的结构为 1-20-1。实验结果为采用两种不同算法的神经网络对 Sinc 数据集训练和测试实验的平均性能,如表 1 所示。

表 1 BPWA 和 LMBP 逼近 Sinc 函数的性能比较

学习算法	训练误差	泛化误差	训练时间/s
BPWA	64.9672	0.35481	36.4992
LMBP	64.5062	0.79746	63.4133

在给出的 Sinc 函数的逼近实验中,BPWA 和 LMBP 几乎具有相同的训练误差,但 BPWA 的泛化误差约为 LMBP 的一半,训练时间更少。

4.2 Hermit 多项式函数的逼近

Hermit 多项式的逼近问题由 Mackay 提出[9],如下所示:

$$F(x) = 1.1(1 - x + 2x^2)\exp\left(-\frac{x^2}{2}\right), \quad x \in R \tag{10}$$

训练样本产生方式如下:样本数 $N = 100$,其中样本输入 x 服从区间[−4,4]内的均匀分布,样本输出为 $F(x_i) + e_i$,e_i 为均值为 0、标准差为 0.1 的正态分布噪声。测试样本为在输入 x 在[−4,4]内等间隔取值所产生的 101 个测试样本对。对该函数逼近实验,网络结构为 1-10-1。实验结果如表 2 所示。

表 2 BPWA 和 LMBP 逼近 Hermit 函数的性能比较

算法	训练误差	泛化误差	训练时间/s
BPWA	0.96476	0.3118	0.052812
LMBP	0.8564	0.494	0.22542

在 Hermit 多项式函数的逼近实验中,BPWA 和 LMBP 训练误差和泛化误差接近,BPWA 的泛化能力略好,但 BPWA 训练时间仅为 LMBP 的 1/4。

4.3 XOR 问题的学习

XOR 问题是神经网络算法测试中最常见的一个问题。选取 2-2-1 的网络结构,权重初值在[−1,1]内随机设定。以 0.01 作为成功学习的误差门限,误差小于门限,认为网络成功训练,可以对 XOR 正确分类;在最大迭代次数内如果误差始终大于门限,则判分类学习失败。实验结果见表 3。

表 3 BPWA 和 LMBP 在 XOR 问题的学习性能比较

算法	正确学习率(%)	学习误差	训练时间/s
BPWA	98%	0.024438	0.23203
LMBP	81%	0.02544	0.79188

从表 3 可见,LMBP 无法在所有情况下对 XOR 成功学习,而 BPWA 几乎在所有的测试中都能成功学习,学习时间约为 LMBP 的 1/3。

通过对比实验,考虑 LM 算法是速度最快和准确度最好的前向网络训练算法,可以认为 BPWA 算法确实具有收敛速度快且能在一定程度上对泛化能力加以保证的特点。

5 结语

提出了一种新颖的前向神经网络学习算法 BPWA。不同于以往的神经网络学习算法,该算法在正反两个过程都能对权值进行调整:在正向阶段按最小范数二乘解原则确定连接隐层与输出层的权值,反向阶段则按误差梯度下降原则调整连接输入层与隐层间的权值。该算法具有很快的训练收敛速度,并且在一定的程度上保证所训练神经网络的泛化能力。通过实验模拟并与 LMBP 算法相比较,展示了新算法的性能。

虽然在文中 BPWA 算法是基于单隐层的三层前向神经网络结构,但其方法可以直接推广为多于三层的前向神经网络的学习算法。

参考文献:

[1] HORNIK K, STINCHCOMBE M, WHITE H. Multilayer feedforward networks are universal approximators[J]. Neural Networks, 1989, 2(5): 359 − 366.

[2] CYBENKO G. Approximation by superpositions of a sigmoidal function[J]. Mathematics Control Signals Systems, 1989, (2): 303 − 314.

[3] JUDD JS. Learning in networks is hard[A]. Proceedings of the 1st IEEE International Conference on Neural Networks[C]. San Diego, CA, 1987. 685 − 692

[4] FREDRIC MH, ROSTANIC I. Principles of Neurocomputing for Science&Engineering[M]. Beijing: China Machine Press, 2003. 120 − 126.

[5] HAGAN MT, MENHAI MB. Training feedforward networks with the Levenberg-Marquardt algorithm[J]. IEEE Trans on Neural Networks, 1994, 5(6): 989 − 993.

[6] 刘惠,袁文燕,姜冬青. 矩阵论及应用[M]. 北京:化学工业出版社, 2003.

[7] HUANG GB, ZHU QY, SIEW CK. Extreme learning machine: a new learning scheme of feedforward neural networks[A]. 2004 International Joint Conference on Neural Networks (IJCNN'2004)[C], 2004.

[8] BARTLETT PL. The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network[J]. IEEE Transactions on Information Theory, 1998, 44(2): 525 − 536.

[9] MACKAY DJC. Bayesian interpolation[J]. Neural Computation, 1992, 4(3): 415 − 447.