第 16 卷 第 2 期　　控 制 与 决 策　　2001 年 3 月
Vol 16 No. 2　　CONTROL AND DECISION　　Mar. 2001

文章编号: 1001-0920(2001)02-0167-05

BP 网络改进算法的性能对比研究

高雪鹏, 丛 爽
(中国科学技术大学 自动化系, 安徽 合肥 230027)

摘 要: 通过实例对几种具有代表性的用以训练BP 网络权值的改进算法进行性能对比研究。首先分析了基于标准梯度下降法和基于标准数值优化方法获得的各种改进算法的优缺点, 然后对各种改进算法在训练中所需的收敛时间及其所达误差进行对比分析。其结果为选择训练网络的算法, 开阔人们对算法改进的思路提供了一些借鉴。

关键词: 梯度下降法; 数值优化; 学习速度; 性能对比

中图分类号: TP 183　　**文献标识码**: A

Comparative Study on Fast Learning Algorithms of BP Networks

GAO Xue-peng, CONG Shuang
(Department of Automation, University of Science and Technology of China, Hefei 230027, China)

Abstract: A comparative study on some typical faster learning algorithms of BP networks is proposed. Firstly, the advantages and disadvantages of two main categories, the fast algorithms based on standard steepest descent and standard numerical optimization, are analysed by means of their work principles. Then the comparative study on convergence times and error performance of training of those faster algorithms is done by two numerical examples. The study can be used for the reference in selecting learning algorithms and developing high performance algorithms.

Key words: gradient descent algorithm; numerical optimization; learning speed; comparative study

1 引　　言

20 世纪 80 年代中期以来, 多层前向网络的反向传播算法(BP 算法)一直备受人们的关注。事实上, 传统的基于标准梯度下降法的BP 算法在求解实际问题时, 常因收敛速度太慢而影响求解质量。为此, 人们在标准BP 算法的基础上进行了许多有益的改进, 提出不少基于非线性优化的训练算法, 可使网络训练的收敛速度比标准梯度下降法快数十乃至数百倍。

本文将改进算法分为两大类: 1) 基于标准梯度下降的改进方法, 此类算法是由标准梯度下降法发展而来的, 其中选用附加动量的BP 算法、学习速率可变的BP 算法和弹性BP 算法进行性能对比研究; 2) 基于标准数值优化的改进方法, 其中选用共轭梯度法、拟牛顿法和 Levenberg-Marquardt 法进行性能对比研究。

本文首先给出各种算法的计算公式及其工作原理, 并从理论上分析了各自的优缺点, 然后通过两个实例详细地进行性能的对比研究, 最后给出了结论。

收稿日期: 2000-04-11; **修回日期**: 2000-06-20
基金项目: 中国科学院盈科优秀青年学者奖项目; 安徽省自然科学基金项目(97413005)
作者简介: 高雪鹏(1976—), 男, 江苏涟水人, 硕士研究生, 从事神经网络建模和模糊建模研究; 丛爽(1961—), 女, 山东文登人, 教授, 博士, 从事人工神经网络和智能控制研究。

2 基于标准梯度下降的方法[1,2]

标准的BP 算法是基于梯度下降法, 通过计算目标函数对网络权值和阈值的梯度进行修正的。改进算法大多是在标准梯度下降法的基础上发展起来的, 它们只用到目标函数对权值和阈值的一阶导数(梯度) 信息。

标准梯度下降法权值和阈值修正的迭代过程可表示为

$$X^{(k+1)} = X^{(k)} - \alpha\nabla f(X^{(k)}) \tag{1}$$

其中, $X^{(k)}$ 为由网络的所有权值和阈值组成的向量, α为学习速率, $f(X^{(k)})$ 为目标函数(采用目标函数而不采用误差函数是因为有时性能指标中含有误差之外的其它项), $\nabla f(X^{(k)})$ 表示目标函数的梯度。

标准BP 算法虽然为训练网络提供了简单而有效的方法, 但由于训练过程中 α 为一较小的常数, 因而存在收敛速度慢和局部极小问题。为解决这些问题, 人们提出许多改进算法, 其中较有代表性的有以下几种。

2.1 附加动量的 BP 算法

附加动量的BP 算法权值修正的迭代过程可表示为

$$X^{(k+1)} = m_c(X^{(k)} - X^{(k-1)}) - (1 - m_c)\alpha\nabla f(X^{(k)}) \tag{2}$$

其中 $0 \le m_c \le 1$ 为动量常数。

附加动量的引入可使网络权值的变化不仅反映局部的梯度信息, 而且反映误差曲面最近的变化趋势。这一算法虽然在一定程度上解决了局部极小问题(对于大多数实际应用问题, 这一能力也极为有限), 但其训练速度仍然很慢。

2.2 学习速率可变的 BP 算法

在标准BP 算法中, 学习速率 α 在训练过程中始终保持恒定。变学习速率的基本思想是: 在保持训练稳定的前提下, 使每次用于修正权值的迭代步长尽可能大。其过程可表示为

$$X^{(k+1)} = X^{(k)} - \alpha^{(k)}\nabla f(X^{(k)}) \tag{3}$$

这一策略在误差增加不太大的范围内, 能提高学习速率, 在局部区域获得一个近最优的学习速率, 从而得到比标准BP 算法更快的收敛速度。然而, 在 $\nabla f(X^{(k)})$ 很小的情况下, 仍然存在权值的修正量很小的问题, 致使学习效率降低。

2.3 弹性 BP 算法

BP 网络通常采用 Sigmoid 隐含层。当输入的 Sigmoid 函数很大时, 斜率接近于零, 这将导致算法中的梯度幅值很小, 可能使对网络权值的修正过程几乎停顿下来。弹性BP 算法只取偏导数的符号, 而不考虑偏导数的幅值。其权值修正的迭代过程可表示为

$$X^{(k+1)} = X^{(k)} - \mathrm{delta}X^{(k)}\,\mathrm{sign}(\nabla f(X^{(k)})) \tag{4}$$

在弹性BP 算法中, 当训练发生振荡时, 权值的变化量将减小; 当在几次迭代过程中权值均朝一个方向变化时, 权值的变化量将增大。因此, 弹性BP 算法的收敛速度要比前几种方法快得多, 而且算法并不复杂, 也不需要消耗更多的内存。

以上 4 种算法的存储量要求相差不大, 而各算法的收敛速度却依次加快。其中, 弹性BP 算法的收敛速度远快于前 3 种。大量实际应用已证明弹性BP 算法是非常有效的。

3 基于数值优化方法的网络训练算法

将上述几种基于一阶梯度的方法用于简单问题时, 往往可以很快地收敛到期望值。然而, 当用于较复杂的实际问题时, 除弹性BP 算法外, 其余算法在收敛速度上都存在一定的问题。

BP 网络的训练实质上是一个非线性目标函数的优化问题。人们对非线性优化问题的研究已有数百年的历史, 而且不少传统数值优化方法收敛也较快, 因而人们自然想到采用基于数值优化的算法对BP 网络的权植进行训练。与梯度下降法不同, 基于数值优化的算法不仅利用了目标函数的一阶导数信息、而且往往利用了目标函数的二阶导数信息[3]。这类算法包括拟牛顿法、Levenberg-Marquardt 法和共轭梯度法, 它们可以统一描述为

$$\begin{cases} f(X^{(k+1)}) = \min\limits_{\alpha} f(X^{(k)} + \alpha^{(k)}S(X^{(k)})) \\ X^{(k+1)} = X^{(k)} + \alpha^{(k)}S(X^{(k)}) \end{cases} \tag{5}$$

其中, $X^{(k)}$ 为网络的所有权值和阈值组成的向量, $S(X^{(k)})$ 为由 X 各分量组成的向量空间的搜索方向, $\alpha^{(k)}$ 为在 $S(X^{(k)})$ 方向上使 $f(X^{(k+1)})$ 达到极小的步长。这样, 网络权值的寻优便可分为以下两步: 1) 首先确定当前迭代的最佳搜索方向; 2) 在此方向上寻求最优迭代步长。下面讨论的 3 种方法, 其区别正在于对最佳搜索方向的选择上有所不同。

3.1 拟牛顿法

牛顿法是一种常见的快速优化方法, 其收敛速度比一阶梯度法快。但由于牛顿法中用到 Hessian

矩阵(二阶导数矩阵), 而导致计算复杂性增加。

为此, 人们在牛顿法的基础上, 提出一类无需计算二阶导数矩阵及其求逆运算的方法。这类方法一般是利用梯度信息或一个近似矩阵去逼近 $H^{(k)}$。比较典型的有 BFGS 拟牛顿法和一步正切拟牛顿法[1]。

3.2 Levenberg-Marquardt 法

众所周知, 梯度下降法在最初几步下降较快, 但随着接近最优值, 由于梯度趋于零, 致使目标函数下降缓慢; 而牛顿法则可在最优值附近产生一个理想的搜索方向。Levenberg-Marquardt 法实际上是梯度下降法和牛顿法的结合, 它的优点在于网络权值数目较少时收敛非常迅速。

3.3 共轭梯度法

梯度下降法收敛速度较慢, 拟牛顿法计算较复杂, 而共轭梯度法则力图避免两者的缺点。共轭梯度法的第一步是沿负梯度方向进行搜索, 然后再沿当前搜索的共轭方向进行搜索, 从而可以迅速达到最优值。

共轭梯度法比大多数常规的梯度下降法收敛快, 并且只需增加很少的存储量和计算量。对于权值很多的网络, 采用共轭梯度法不失为一种较好的选择。

综上所述, 考虑到算法所需的存储量和收敛速度, 在选择算法对网络进行训练时, 可以遵循以下原则:

1) 当网络参数很少时, 可选用牛顿法或 Levenberg-Marquardt 法;

2) 当网络参数适中时, 可选用拟牛顿法;

3) 当网络参数很多时, 考虑到存储容量问题, 不妨选择共轭梯度法。

需要指出的是, 本文述及的所有算法均存在局部极小问题。就经验而言, 对大多数问题, Levenberg-Marquardt 法可获得较好的结果。然而, 这一结论并没有任何理论依据, 所以对网络应使用不同的初始值进行多次训练。此外, 还可采用其它全局优化算法(如模拟退火法和遗传算法等) 来解决局部极小问题。

4 数值实验

本节给出两个例子, 均在MATLAB5.3 环境下进行。第 1 个例子是利用神经网络的非线性逼近特性对一个非线性函数对象进行逼近; 第 2 个例子是利用神经网络对一个真实的非线性直流电机进行建模。

4.1 非线性函数的逼近

构造非线性函数对象[4] 为

$$y(k) = \frac{y(k-1)}{1+y^2(k-1)} + u^3(k-1) \qquad (6)$$

训练用输入信号取为

$$u(k) = 0.2\sin\frac{2\pi k}{25} + 0.3\sin\frac{\pi k}{15} + 0.3\sin\frac{\pi k}{75} \qquad (7)$$

将信号(7) 作为非线性对象(6) 的输入信号, 取 k 从 0 到 200 的输入/输出作为训练样本。网络训练样本的输入/输出关系如图 1 所示。

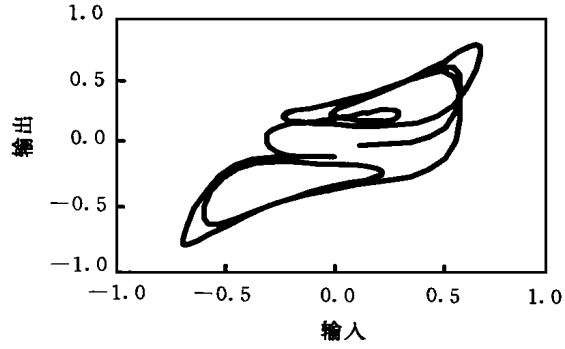

图 1　输入/输出关系曲线

在MATLAB5.3 中, 如果希望一个网络完成某种功能, 则可通过调用函数 newff 实现网络的创建, 然后调用函数 train 对所建网络 newff 进行训练。

函数newff 带有 4 个参数。对于一个输入节点数为 R 的网络, 第 1 个参数是一个 $R*2$ 的矩阵, 其中, 第 i 行的两个元素依次为所有输入样本中对应于第 i 个输入的最小和最大范围; 第 2 个参数是给出各层神经元个数的数组; 第 3 个参数是各层采用的激活函数的名称; 第 4 个参数是所选用的网络训练算法的名称。newff 的返回值是一个经过初始化的网络对象。本例中, 用于对非线性函数逼近的网络可创建如下

$$\begin{cases} p = [u_d, y_d] \\ \text{net} = \text{newff}(\text{minmax}(p), [5,1], \\ \qquad \{'\text{tansig}','\text{purelin}'\}, '\text{traingd}') \end{cases} \qquad (8)$$

其中, u_d 和 y_d 分别为输入和输出信号的一阶延时构成的向量。在神经网络工具箱中, 前述各种训练算法已编制成•m 文件, 设计者只需在 newff 的第 4 个参数中选用希望采用的算法名称, 便可调用函数 train 对所建网络进行训练。函数 train 有 3 个参数: 第 1 个参数是由newff 创建的网络名称; 第 2 个参数是输入

矩阵; 第3个参数是期望输出矩阵。

$$net = train(net, p, y) \tag{9}$$

在式(8)和(9)中, u_d 和 y_d 组成输入矩阵 p, 网络的期望输出为 y, minmax(p) 为取得输入向量的最小到最大的范围。网络的第1层(即隐含层)含有5个节点, 第2层(即输出层)含有1个节点; 隐含层采用 tan-Sigmoid 函数, 输出层采用线性 purelin 函数。所选训练算法 traingd 为标准的梯度下降法。

作为对比研究, 我们始终采用式(8)中的网络结构(只改变最后的算法参数), 并采用式(9)对网络进行训练, 以此来观察各种训练算法的收敛速度。

训练中, 以固定最大迭代次数20 000为标准。网络的初始化采用 initnw 初始化函数, 它可使每一层神经元的激活区域大致均匀地分布在输入空间。表1给出了3种算法收敛速度的比较。表中的数据均为5次训练的平均值。

表1 3种算法的收敛速度比较

函 数	算法描述	时间(s)	均方误差
traingda	变学习速率的BP算法	448.30	0.000 265 96
traingdm	附加动量的BP算法(动量因子 $m_c = 0.9$)	457.75	0.002 190 00
traingd	标准梯度下降法(标准BP算法)	446.38	0.003 834 69

由表1可见, 3种方法所用时间均在400s以上。在误差的变化方面, 学习速率可变的BP算法的收敛速度比其它两种快得多, 已达到0.000 1的数量级。

鉴于弹性BP算法和基于数值优化算法的收敛速度很快, 我们用固定目标函数(均方误差)MSE = 0.000 1来进行性能的对比实验, 使各算法一直迭代到满足目标要求为止。5次成功训练的平均值如表2所示。表中给出了MATLAB5.3中所有4种不同的共轭梯度法和2种不同的拟牛顿法的训练结果。

不难看出, 表2给出的8种算法无论是运算时间, 还是收敛精度, 均比表1中的3种算法优越得多, 二者不在同一个数量级上。由此可见, 在标准梯度下降法基础上改进的算法, 所取得的进步是很有限的。要想在数量级上有所突破, 必须转变思路。基于数值优化的方法便是一个有益的尝试。在今后的研究与应用中, 应采用这些现成软件编制好的训练网络算法, 以提高网络设计的精度, 灵活地应用于实际问题。

4.2 利用神经网络逼近非线性直流电机的输入/输出特性

下面对具有非线性摩擦力影响的直流电机的输入/输出特性进行神经网络建模, 并采用不同的训练算法进行对比。实际控制系统包括一台Pentium 200微机, 一块内置于计算机的12位A/D和D/A转换板, PWM功率放大电路, 直流力矩电动机及直流测速发电机。模拟电压输入范围和输出控制电压范围均为[-5, 5]V, 模数转换后的数字量范围均为[-2 048, 2 048]。为方便起见, 输入和输出均采用数字量。

给电机系统输入幅值为300, 周期为10s的正弦信号, 在5ms的采样周期下, 获得电机的真实输出(即转速信号)。实际电机的输入/输出如图2所示。由图可以看出, 系统存在严重的非线性特性。

假设电机模型为一阶系统, 即输入/输出关系为

$$y(k) = f(y(k-1), u(k-1)) \tag{10}$$

采用输入和输出的一阶延时作为网络的输入, 在目标均方误差MSE = 2的条件下训练网络。仍采用例1的网络结构newff训练网络模型, 训练集样本数为2 000个。

表2 快速训练算法的收敛速度对比

函 数	算法描述	时间(s)	迭代次数
traincgf	Fletcher-Powell 共轭梯度法, $\beta^{(k)} = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}}$	14.34	238
traincgb	Polak-Ribiere 共轭梯度法, $\beta^{(k)} = \frac{\Delta g_{k-1}^T g_k}{g_{k-1}^T g_{k-1}}$	12.32	216
traincgb	Powell-Beale 共轭梯度法	10.11	167
trainscg	Scaled 共轭梯度法	11.76	263
trainbfg	BFGs 拟牛顿法	4.34	71
trainoss	一步割线拟牛顿法	22.50	437
trainlm	Levenberg-Marquardt 法	0.77	7
trainrp	弹性BP算法	50.20	2 239

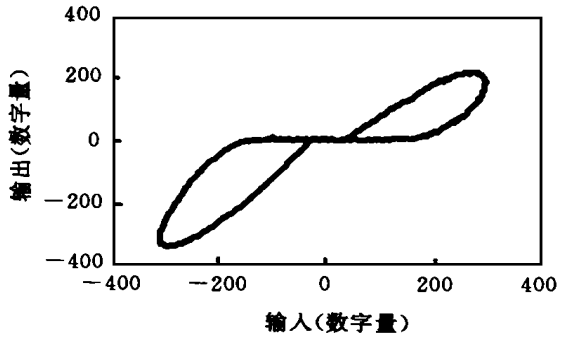

图 2　实际电机输入／输出关系

实验表明, 采用标准梯度下降法、附加动量的 BP 算法和学习速率可变的BP 算法训练网络模型, 将无法达到性能目标。我们曾采用可变学习速率的 BP 算法训练了 3 小时, 训练的误差离目标相差仍甚远。

经过多次实验, 除 trainscg 共轭梯度法外, 其余 3 种共轭梯度法均在离目标较远的点陷入局部极小值, 使得搜索步长趋近于零而终止训练。虽然 trainscg 法没有陷入局部极小值, 但收敛也很慢。曾进行 100 000 次迭代, 耗时近 3 小时, 均方误差MSE 才达到 22 829 5(考虑到输出信号的幅值, 这个结果勉强可以接受)。拟牛顿法中的 trainoss 法收敛速度也很慢, 根本无法使用。

最后, 只有表 3 给出的 3 种方法能较快地达到训练要求。由此可见, 在实际应用中, 应针对具体问题尝试多种算法训练网络, 以达到最佳效果。另外, 从两个例子还看出,BFGs 拟牛顿法和 Levenberg-Marquardt法均有上乘表现, 而后者的收敛速度最快, 在相同的误差目标下所花费的时间之短, 是其它算法无法比拟的。

表 3　3 种算法的比较

函　数	算法描述	时间 (s)	迭代次数
trainrp	弹性BP 算法	314 12	3 483
trainbfg	BFGs 拟牛顿法	29 50	133
trainlm	Levenberg-Marquardt 法	10 27	45

5　结　　语

本文对几种训练BP 网络权值的改进算法进行了性能上的对比研究。研究表明, 无论是用于非线性函数逼近还是用于非线性系统建模, 基于标准数值优化方法的各种改进算法, 均比基于标准梯度下降法的改进算法在收敛速度方面提高一个数量级以上。这便为采用神经网络对复杂系统进行建模与控制提供了可行的依据。尽管数值优化法运用起来较为复杂, 但在软件工具(如MATLAB 神经网络工具箱)的帮助下, 只要进行函数的简单调用即可。然而, 我们所研究的算法均存在陷入局部极小值的可能性, 这一问题可通过其它途径加以解决。

参考文献:

[1] Howard Demuth, Mark Beale. Neural network toolbox user's guide for use with MATLAB (Version 5. 3) [M]. Natic Mass: The MathWorks, Inc, 1998.

[2] Sarle W S. Neural network FAQ — Part 2 of 7: Learning [M]. Cary NC: Periodic Posting to the Usenet Newsgroup Comp ai Neuralnets, 1999.

[3] 孙德敏. 工程最优化方法及应用[M]. 合肥: 中国科技大学出版社, 1997.

[4] Sangbong Park, Cheol Hoon Park. Adaptive system identification using multilayer neural network and Gaussian potential function networks [A]. The 1996 IEEE Int Conf on Neural Networks [C]. Piscataway: IEEE, 1996. 4: 2261-2265.

[5] 丛爽. 面向MATLAB 工具箱的神经网络理论与应用[M]. 合肥: 中国科技大学出版社, 1998.