基于改进共轭梯度法的前馈网络快速监督学习算法

杨 斌[1],聂在平[1],夏耀先[2],蒋荣生[2]

(1. 电子科技大学电子工程学院,四川成都 610054;2. 中海油田服务有限公司,北京 101149)

摘 要: 为了提高多层前馈神经网络的权参数的学习效率,通过引入改进的求解大规模线性方程组的共轭梯度法,提出一种新的基于 LM 的前馈网络学习算法.该算法不仅具有 LM 优化学习方法的快速收敛特性,而且降低了 LM 法的计算复杂度,可获得比其它标准算法更好的学习精度和推广预测能力.文中通过仿真结果证明了新算法在函数逼近和时间序列预测等问题环境下的有效性.

关键词: 神经网络; Levenberg-Marquardt 算法; 共轭梯度法; 监督学习

中图分类号: TP183 **文献标识码:** A **文章编号:** 0372-2112 (2002) 12-1845-03

A Fast Supervised Learning Algorithm of Multilayer Feedforward Network Based on Improved Conjugate Gradient Method

YANG Bin[1], NIE Zai-ping[1], XIA Yao-xian[2], JIANG Rong-sheng[2]

(1. *Dept. of Electronic Engineering, UEST of China, Chengdu, Sichuan* 610054, *China*;
2. *China National Offshore Oil Corporation, Service, Beijing* 101149, *China*)

Abstract: To improve the weight learning efficiency of multilayer feedforward network, a new similar LM learning algorithm is proposed by introducing modified conjugate gradient method in solving of large-scale linear equation sets. In addition to the fast convergent advantage the LM method demonstrates that, the new algorithm not only reduces the training time and overall complexity, but also achieves training accuracy and generalization capability comparable to more standard approaches. Extensive simulation results are provided to show the effectiveness of the new algorithm.

Key words: neural network; Levenberg-Marquardt algorithm; conjugate gradient method; supervised learning

1 引言

神经网络是基于"从样本中学习"的技术去完成对一个未知复杂函数 F 的近似.当把网络的训练学习看作是一个非线性最优化问题时,对目标函数 E 的最小化就可通过应用非线性最优化领域中的各种方法[1,2]来完成.基于二阶法[3]的最优化方法,由于具有很好的收敛特性,近年来被应用于神经网络的学习训练中.其思想是利用误差函数对被优化权参数的二阶导数矩阵——Hessian 矩阵的信息或者计算 Hessian 矩阵的近似矩阵.但越来越多的分析结论指出[4]神经网络学习中获得的 Hessian 矩阵或用于近似 Hessian 的 $\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}$ 矩阵($\boldsymbol{J}$ 为 Jacobian 矩阵)由于 Sigmoid 等单元传输函数的饱和特性、输出单元间的线性依赖性及局部梯度向量间的线性依赖性而变得病态,会带来结果解的不稳定性及较大的条件数情况下十分缓慢的迭代收敛或收敛失败.使用 Levenberg-Marquardt 方法(简称 *LM* 法)能克服 Hessian 矩阵的病态和奇异性而且可给出稳定的解和快的收敛速度,已被成功地应用于神经网络学习训练中[4~6].但是,目前的 LM 学习方法为求解线性方程而需很大的内存空间及较高的计算代价(至少是 $O(N^3/6)$的计算复杂度),尤其是当网络规模增大到成百上千个权参数或训练样本数增多时,其快速收敛性被高昂的计算代价所消减,失去了原有的优于 BP 等算法的特性,在实际应用中受到了限制,本文就是针对基于 LM 的前馈神经网络学习算法的改进.

2 基于 LM 的前馈网络训练算法

对给定的训练数据集$\{(\boldsymbol{x}_i,\boldsymbol{t}_i)\}_{i=1}^{M}$,前馈网络监督学习的目标就是通过迭代式地调节网络连接权向量 $\boldsymbol{W}$(设有 N 个分量)使得特定的累积误差平方和目标函数 $E(\boldsymbol{W})=\frac{1}{2}\boldsymbol{e}^{\mathrm{T}}(\boldsymbol{W})\boldsymbol{e}(\boldsymbol{W})$被最小化而获得模型,完成对学习样本点的拟合任务.这里 $\boldsymbol{e}$ 为样本期望输出与网络实际输出之差值的误差向量($e_i=y_i-t_i$).把对 $E(\boldsymbol{W})$的最小化看作一个非线性无约束最小二乘问题,可利用对高斯-牛顿法加以改进的 LM 优化更新规则:

$$\min \| \boldsymbol{J}\Delta\boldsymbol{W}+\boldsymbol{e} \| \tag{1}$$

$$(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}+\mu\boldsymbol{I})\Delta\boldsymbol{W}=-\boldsymbol{J}^{\mathrm{T}}\boldsymbol{e} \tag{2}$$

$$\boldsymbol{W}^{(k+1)}=\boldsymbol{W}^{(k)}+\Delta\boldsymbol{W} \tag{3}$$

而得到网络权向量 $\boldsymbol{W}^{(k+1)}$.其中 k 为迭代步数,$\|\cdot\|$ 为 Euclidean 模,$\boldsymbol{J}=[\partial \boldsymbol{e}_i/\partial \boldsymbol{W}_j]_{M\times N}$是对应于多层前馈网络的 Jacobian 矩阵,$\boldsymbol{I}$ 为单位矩阵.μ 一般被称为阻尼因子,它的作用类似于牛顿步长,是一个需要在迭代过程中调整的重要参数,较为复杂的选择方法是基于信赖域模型法[1,2]和线性搜索法[1].它们都试图确定一个用二次函数足以逼近非线性问题内的区域以保证算法对目标函数的局部极小的收敛性.信赖域模型法通过目标函数的实际减小与预测减小的比值来对

收稿日期:2001-09-24;修回日期:2001-01-24
基金项目:国家自然科学基金(No.69871004);国土资源部油气藏地质与开发工程国家重点实验室基金(No.PLC9913)

阻尼因子增大或减小,线性搜索法使用一个类似的线性化程度准则来检查 μ_k 是否满足该准则,若不满足则通过增大 μ_k 值直至满足为止.它们的计算细节、计算复杂度及应用性能对比将另文探讨.此处使用了一个简单有效的方法,即依据当前步目标函数值 $E_{k+1}(\boldsymbol{W})$与上一步目标函数值 $E_k(\boldsymbol{W})$的相对大小来调整阻尼因子参数:若 $E_{k+1}(\boldsymbol{W}) > E_k(\boldsymbol{W})$,则增大 μ_k,令 $\mu_k = \mu_k * r_1(r_1 > 1.0)$;否则若 $E_{k+1}(\boldsymbol{W}) \leq E_k(\boldsymbol{W})$,则减小 μ_k,令 $\mu_k = \mu_k / r_2(r_2 > 1.0)$.比例因子 r_1(缺省 = 10.0)和 r_2(缺省 = 2.0)在程序运行起始处人为设定.

3 改进的共轭梯度法

实际上,对式(2)法方程组的求解方法的选择是决定 LM 学习算法计算复杂性及是否高效的核心问题.已有很多求解式(2)的方法,如 Cholesky 分解法[6]、三角分解法[4]、LU 分解法[7]和 QR 分解法[7]等,但这些方法的计算复杂度都较高($O(N^3/6)$),尤其对于变量数或数据样本较多的实际应用问题,在计算速度、结果稳定性、抗病态性上均难以达到要求,限制了 LM 法的应用规模和范围.共轭梯度法目前被一致认为是求解大型线性方程组最有效、最快速的方法之一.对于式(2)的求解,可以借助于共轭梯度法思想来加以推广,为此我们提出了用改进的共轭梯度法(简称 DCG)求解 LM 阻尼正则方程组的快速算法.这里,令 $\boldsymbol{B}$ 为式(1)中的残差向量 $\boldsymbol{e}^{\mathrm{T}}(\boldsymbol{W})$.在第 k 个迭代步内,$\Delta\boldsymbol{W}^{(k)}$为当前迭代步的权更新解向量,$\boldsymbol{P}^{(k)}$为方向向量,$\boldsymbol{H}^{(k)}$为共轭梯度向量,$\boldsymbol{R}^{(k)}$为余量向量,$\boldsymbol{D}$ 为阻尼因子向量($\boldsymbol{d}_i = \mu$),$\boldsymbol{Q}^{(k)}$和 $\boldsymbol{V}^{(k)}$为临时向量,$\boldsymbol{Q}^{(k)} = \boldsymbol{D}\boldsymbol{P}^{(k)}$,$\boldsymbol{V}^{(k)} = \boldsymbol{D}\Delta\boldsymbol{W}^{(k)}$.$\boldsymbol{J}$ 为 $M \times N$ 维的 Jacobian 矩阵,α_k 和 β_{k+1}均为标量,分别为向量 $\Delta\boldsymbol{W}^{(k)}$和 $\boldsymbol{P}^{(k)}$的修正因子或修正步长.(·,·)表示内积运算.具体算法步骤为:

(1)设置最大迭代步数 *KMAX* 和误差精度 ε.令 $k = 0$,初始化向量 $\Delta\boldsymbol{W}^{(0)}$、$\boldsymbol{H}^{(0)}$、$\boldsymbol{V}^{(0)}$、$\boldsymbol{P}^{(0)}$和 $\boldsymbol{R}^{(0)}$,其中 $\boldsymbol{H}^{(0)} = \boldsymbol{B} - \boldsymbol{J}^{\mathrm{T}}\Delta\boldsymbol{W}^{(0)}$,$\boldsymbol{V}^{(0)} = \boldsymbol{D}\Delta\boldsymbol{W}^{(0)}$,$\boldsymbol{P}^{(0)} = \boldsymbol{R}^{(0)} = \boldsymbol{J}^{\mathrm{T}}\boldsymbol{H}^{(0)} - \boldsymbol{V}^{(0)}$.

(2)计算 $\alpha_k = (\boldsymbol{R}^{(k)}, \boldsymbol{R}^{(k)})/((\boldsymbol{J}\boldsymbol{P}^{(k)}, \boldsymbol{J}\boldsymbol{P}^{(k)}) + (\boldsymbol{P}^{(k)}, \boldsymbol{Q}^{(k)}))$.

(3)调整更新解向量 $\Delta\boldsymbol{W}^{(k+1)} = \Delta\boldsymbol{W}^{(k)} + \alpha_k\boldsymbol{P}^{(k)}$.

(4)计算 $\boldsymbol{H}^{(k+1)} = \boldsymbol{H}^{(k)} - \alpha_k\boldsymbol{J}\boldsymbol{P}^{(k)}$和 $\boldsymbol{R}^{(k+1)} = \boldsymbol{J}^{\mathrm{T}}\boldsymbol{H}^{(k+1)} - \boldsymbol{V}^{(k+1)}$.若 $\| \boldsymbol{R}^{(k+1)} \| \leq \varepsilon$,则输出 $\Delta\boldsymbol{W}^{(k+1)}$作为方程组(5)的解并停止迭代.否则,继续以下步骤.

(5)计算修正步长 $\beta_{k+1} = (\boldsymbol{R}^{(k+1)}, \boldsymbol{R}^{(k+1)})/(\boldsymbol{R}^{(k)}, \boldsymbol{R}^{(k)})$.

(6)修正方向向量 $\boldsymbol{P}^{(k+1)} = \boldsymbol{R}^{(k+1)} + \beta_{k+1}\boldsymbol{P}^{(k)}$.

(7)若 $k > KMAX$,则停止迭代并输出 $\Delta\boldsymbol{W}^{(k+1)}$,否则,令 $k = k + 1$,去步骤 2 继续.

从上述改进的共轭梯度算法公式可以看出,与求解线性方程组的标准共轭梯度法[7](CG)相比,在每个迭代步 DCG 仅需多做一次矩阵与向量的乘法,这部分总计算量(乘法次数)是 $2N^2$,其余的计算量都是 N 的线性函数,因此,每次 DCG 的运算量是 $O(2N^2)$.

表 1 5 个函数的学习和预测性能结果对比

性能	算法	Sif 函数	Rad 函数	Harm 函数	Cadd 函数	Cif 函数
迭代次数	BFGS	191(83.7)	83(24.2)	315(220.2)	255(111.3)	327(174.5)
	SCG	436(188.7)	138(57.3)	533(239.9)	448(224.8)	677(332.6)
	PRCG	444(186.4)	142(48.4)	528(177.5)	539(242.7)	739(372.6)
	LMMLP	56(28.2)	33(10.7)	141(130.9)	104(42.0)	138(183.4)
	DCGLM	53(13.1)	28(8.1)	98(34.9)	48(25.3)	115(71.2)
FVU 训练误差	BFGS	0.0587(0.0001)	0.0588(0.0003)	0.1021(0.0005)	0.0624(0.0002)	0.0676(0.0002)
	SCG	0.0587(0.0009)	0.0589(0.0002)	0.1024(0.0002)	0.0625(0.0001)	0.0677(0.0001)
	PRCG	0.0587(0.0001)	0.0589(0.0003)	0.1022(0.0004)	0.0624(0.0001)	0.0681(0.0017)
	LMMLP	0.0584(0.0005)	0.0564(0.0024)	0.1003(0.0025)	0.0616(0.0014)	0.0672(0.0008)
	DCGLM	0.0425(0.0003)	0.0397(0.0041)	0.0715(0.0281)	0.0405(0.0070)	0.0503(0.0062)
FVU 检验误差	BFGS	0.0098(0.0027)	0.0223(0.0031)	0.1483(0.0489)	0.0412(0.0081)	0.0545(0.0105)
	SCG	0.0110(0.0032)	0.0234(0.0056)	0.1433(0.0389)	0.0430(0.0064)	0.0622(0.0210)
	PRCG	0.0112(0.0032)	0.0229(0.0055)	0.1636(0.0512)	0.0432(0.0098)	0.0591(0.0161)
	LMMLP	0.0093(0.0029)	0.0180(0.0029)	0.1392(0.0499)	0.0383(0.0098)	0.0532(0.0083)
	DCGLM	0.0044(0.0017)	0.0133(0.0029)	0.1012(0.0573)	0.0333(0.0120)	0.0487(0.0099)

4 实验模拟结果

4.1 5 个二维非线性函数回归问题

这里使用了来自文[8]的 5 个二维非线性函数的学习来说明本文算法(DCGLM)在函数逼近上的性能.这 5 个函数分别是 Sif(Simple interaction function)、Rad(Radial function)、Harm(Harmonic function)、Cadd(Additive function)、Cif(Complicate interaction function).其具体形式及生成各自 225 个含噪声训练数据及 10000 个检验用数据的方法参见文[8].预测误差测度使用 FVU(Fraction of Variance Unexplained)值,其定义为:FVU = $\sum_{i=1}^{M}(t(\boldsymbol{x}_i) - y(\boldsymbol{x}_i))^2 / \sum_{i=1}^{M}(t(\boldsymbol{x}_i) - \bar{t})^2$,$M$ 为样本总数,$\boldsymbol{x}_i$ 为第 i 个样本的输入向量,$t(\boldsymbol{x}_i)$为对应的期望输出函数值,$y(\boldsymbol{x}_i)$为网络实际输出值,$\bar{t} = (1/M)\sum_{i=1}^{M} t(\boldsymbol{x}_i)$.网络结构为 2 - 10 - 1,10 个隐单元用 Sigmoid 传输函数,输出单元为线性函数.使用基于 LU 分解[8]的 LM 学习算法(LMMLP)、带有三次插值线性搜索的拟牛顿优化法[1](BFGS)、Polak-Ribere 共轭梯度法[2](PRCG)、Moller[9]的自适应学习步长共轭梯度法(SCG)等 4 个算法对这 225 个含噪声数据的训练及预测结果如表 1 所示.

其中每个算法用不同的初始权值独立运行 60 次而得到 FVU 误差的均值和标准偏差(括号内),FVU 训练误差是在学习数据上得到的误差,FVU 检验误差是在检验数据集上得到的误差,训练停止准则为均方误差 MSE 小于 0.11(对 Harm 函数)或 0.07～0.06(对其它函数).可见,我们的新算法在学习收敛速度和预测推广能力上均有较好的表现.对于 Sif、Rad 函数,其二维曲面形状较简单,易于学习,DCGLM 与 LMMLP 二者收敛速度相近,使用了最少的迭代步数.对于另外三个较复杂的二维函数,则需更多的迭代步.BFGS、PRCG 法均需要进行繁重的常规一维搜索,SCG 使用单边微分方程来寻找最优单步线性步长,都额外增加了计算代价.DCGLM 和 LMMLP 均利用了二阶信息,不需线性搜索步长,迭代速度要快的多,而且 DCGLM 比 LMMLP 收敛更快.尤其突出的是,DCGLM 在 5 个函数上均获得了最小的推广误差.

4.2 Building 时间序列预测问题

这是一个来自 Prechelt[10] 编辑的实际问题数据集.任务是根据连续 4 个月中在每个小时上的 14 个观测属性数据来预测后续的 2 个月内每个小时上的耗电量、热水和冷水消耗量.这是一个时间序列的外推预测问题.使用平方误差百分比 SEP 作为误差测度:$SEP = 100.0 \times \frac{O_{max} - O_{min}}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (O_{ij} - T_{ij})^2$,$O_{max}$ 和 O_{min} 分别为输出元的最大、最小网络输出值,O_{ij} 为网络的实际输出值,N 为输出层单元数,M 为样本数,T_{ij} 为样本目标输出值.使用 14-10-3 网络,10 个隐元均为双曲正切传输函数,训练停止准则为早期停止技术.用[－1,1]间不同的初始权值对上述 5 个算法分别独立运行 60 次获得了在训练集和检验集上 SEP 误差的均值、标准方差结果列于表 2 中.其中还有来自 Prechelt[10] 用 RPROP[11](弹性 BP)算法在此问题上给出的测试结果.可见,与 RPROP 算法及常规的 SCG、PRCG、BFGS、LMMLP 等算法相比,DCGLM 在 Building 问题上具有最快的学习训练速度和最小的预测误差.

表 2　Building 问题学习及检验预测结果

算　法	网络结构	总迭代次数		训练集误差		验证集误差		检验集误差	
		均值	标准偏差	均值	标准偏差	均值	标准偏差	均值	标准偏差
RPROP	14－16－3	307	544.0	0.4700	0.2800	2.0700	1.0400	1.3600	0.6300
BFGS	14－10－3	392	49.9	0.0998	0.0002	1.2307	0.3646	0.7258	0.0525
SCG	14－10－3	860	179.2	0.1000	0.0003	1.0517	0.2680	0.7319	0.0389
PRCG	14－10－3	956	183.9	0.1001	0.0006	0.9457	0.1401	0.7294	0.0400
LMMLP	14－10－3	113	22.5	0.0924	0.0047	0.8914	0.0902	0.6927	0.0482
DCGLM	14－10－3	50	12.5	0.1136	0.0567	0.8124	0.0752	0.6191	0.0909

5　结论

本文是在基于 LM 的二阶前馈网络学习训练算法的基础上,针对其较高的计算复杂度、较大的内存占用量等缺陷,提出了一种新的学习算法,主要是使用改进的共轭梯度法来求解非线性法方程组,在保持二阶梯度学习算法的快速收敛和不需一维线性搜索等优点的同时,降低了计算复杂度,减少了内存占用,在计算量、适应性方面优于 LM 及基于 BP 的各种改进梯度算法.需进一步研究的是阻尼因子的最优选取等问题并做更多的实际应用.另外,本文的方法并不限于前馈网络的学习训练中,而具有一定的普遍应用意义.

参考文献:

[1] J E Dennis, R B Schnabel. Numerical Methods for Unconstrained Optimization and Nonlinear Equations[M]. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[2] 袁亚湘,孙文瑜.最优化理论与方法[M].北京:科学出版社,1997.183－403.

[3] R Battiti. First-and second order methods for learning: Between steepest descent and Newton's method[J]. Neural Computation, 1992, 4(2): 141－166.

[4] G Zhou, J Si. Advanced neural-network training algorithm with reduced complexity based on Jacobian deficiency[J]. IEEE Trans Neural Networks, 1998, 9(3): 448－453.

[5] S Kollias, D Anastassiou. An adaptive least squares algorithm for the efficient training of artificial neural networks[J]. IEEE Trans Circ Syst, 1989, 36(8): 1092－1101.

[6] M T Hagan, M B Menhaj. Training feedforward networks with the marquardt algorithm[J]. IEEE Trans Neural Networks, 1994, 5(6): 989－993.

[7] P E Gloub, C F Van Loan. Matrix Computations (2nd ed)[M]. Baltimore: Johns Hopkins University Press, 1989.

[8] Hwang J, Lay S, Maechler M. Regression modeling in back-propagation and projection pursuit learning[J]. IEEE Trans Neural Networks, 1994, 5(3): 342－353.

[9] M Moller. A scaled conjugate gradient algorithm for fast supervised learning[J]. Neural Networks, 1993, 6(4): 525－533.

[10] L Prechelt. A quantitative study of experimental evaluation of neural network learning algorithms[J]. Neural Networks, 1996, 9(4): 457－462.

[11] M Riedmiller, H Braum. A direct adaptive method for faster backpropagation learning: the RPROP algorithm[A]. Proc of the IEEE Int Conf on Neural networks[C]. San Francisco, CA: IEEE, April 1993.

作者简介:

杨　斌　男,1967 年出生于四川,副教授,1999 年获成都理工大学博士学位,现在电子科技大学从事博士后研究,研究领域包括:神经网络、智能信息处理、地球物理反演等.