

基于样本期望训练数的 BP 神经网络改进研究

李志俊,程家兴,金奎,饶玉佳

(安徽大学 计算智能与信号处理教育部重点实验室,安徽 合肥 230039)

摘要:BP 算法是神经网络中最常用的算法之一。标准 BP 算法存在的最大问题就是易于陷入局部极小、收敛速度慢等问题。针对 BP 算法的这些问题,出现了许多改进的措施,如引入变步长法、加动量项法等。提出了一种基于样本期望训练数的改进 BP 算法,仿真实验说明了该算法可以明显提高 BP 网络学习速度,并且具有简单通用性,可以和其他方法结合,进一步提高算法的收敛速度。

关键词:神经网络; BP 算法; 样本期望训练数; 收敛速度

中图分类号:TP183

文献标识码:A

文章编号:1673-629X(2009)05-0103-04

BP Algorithm Improvement Based on Sample Expected Training Number

LI Zhi-jun, CHENG Jia-xing, JIN Kui, RAO Yu-jia

(Ministry of Education Key Laboratory of Intelligent Computing &
Signal Processing, Anhui University, Hefei 230039, China)

Abstract: BP algorithm is one of the most widely used algorithms in neural network. In view of the main limitations of BP algorithm, such as easy to fall into local minimum value and slow in convergence, several methods such as momentum term and variable step-size etc. are led to optimize BP algorithm. Proposed a new kind of improvement BP algorithm based on sample expected training number. Experiment results show that this algorithm has superiority in convergent velocity, simplicity and commonality.

Key words: neural network; BP algorithm; sample expected training number; convergent velocity

0 引言

在 20 世纪 80 年代中期,美国学者 Rumelhart^[1]和 McClelland 领导的科学家小组完整地提出了误差反向传播的学习算法,即 BP 算法,系统地解决了多层前馈神经网络的权值调整的学习问题。目前,这个神经网络的模型已经广泛应用于图像处理、模式识别、自动控制、故障诊断、数据预测和函数逼近等各个方面,它的理论发展也是日趋成熟。虽然 BP 网络在现有的神经网络应用中占了相当大的比重,但由于 BP 算法的误差函数曲面高度复杂,并且按梯度下降法调整权值,使得网络在学习过程中出现了收敛速度慢、易陷入局部极小点等缺点^[2~4]。因此如何提高 BP 算法的学习速度是人们一直关注的问题,很多的研究者从各个不同的角度出发对 BP 算法作了大量的改进研究,提出了

很多解决的方法。如早期提出的引入动量项、从固定学习步长到可变学习步长、寻求新的激活函数以取代传统的 sigmoid 函数,采用遗传算法优化等^[5~8]。从心理学中的记忆遗忘规律出发,提出了一种基于样本期望训练数的改进 BP 算法。通过几个计算实例,说明了该算法的可行性和有效性。

1 BP 算法

BP 人工神经网络一般采用三层网络结构,包括输入层、隐含层和输出层,相邻层之间通过权值实现联结,其神经网络的结构如图 1 所示。文中论述的算法都是以三层网络结构出发进行讨论的。

BP 算法的基本思想归结如下:

学习过程由信号的正向传播与误差的反向传播两个过程组成。正向传播时, BP 网络对于输入信号,通过输入层传播到隐含层,经过激励函数的作用,再将隐含层的输出作为输出层的输入传播到输出层,最后得到输出结果。对于每一个输入样本,神经网络的实际输出与目标输出不可能完全一样,两者之间必然会有误差,定义误差为 $E = \frac{1}{2} \sum_{k=1}^L (d_k - o_k)^2$, 其中 L

收稿日期:2008-09-11

基金项目:国家自然科学基金(60273043);教育部博士点基金(200403057002);安徽省自然科学基金(050420204)

作者简介:李志俊(1977-),男,硕士研究生,主要研究方向为智能计算及最优化方法;程家兴,教授,博导,主要研究方向为智能计算及最优化方法。

为输出结点数, d_k 和 o_k 分别为网络的理想输出和实际输出。

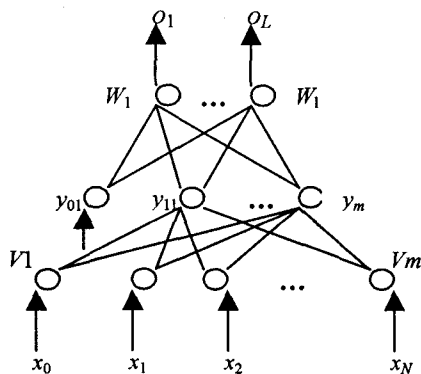


图 1 三层 BP 网络

转入误差的反向传播阶段。误差的反向传播是将输出误差以某种形式通过隐层向输入层逐层反传,并将误差分摊给各层的所有单元,从而获得各层单元的误差信号,此误差信号即作为修正各单元权值的依据。BP 算法按照误差减少最快的方向改变各层间的联结权值,使网络慢慢收敛。取联结权值的增量变化为 $\Delta w_{jk} = -\eta \frac{\partial E}{\partial w_{jk}}$, $\Delta v_{ij} = -\eta \frac{\partial E}{\partial v_{ij}}$, 其中 W, V 为网络的权值矩阵。

这种信号正向传播与误差反向传播的各层权值调整过程,是周而复始地进行的。权值不断调整的过程,也是网络的学习训练过程。此过程一直进行到网络输出的误差减少到可接受的程度,或进行到预先设定的学习次数为止^[3,9]。

2 BP 算法的改进

传统 BP 算法存在的最主要问题就是易于陷入局部极小、收敛速度慢等问题。针对 BP 算法的这些问题,出现了许多改进的措施,如引入变步长法、加动量项法、引入陡度因子等。这些方法一定程度上加快了 BP 算法的收敛速度。在分析 BP 算法原理的基础上,从心理学中的记忆遗忘规律出发,提出了一种基于样本期望训练数的改进 BP 算法。

2.1 样本的期望训练数

BP 算法的训练过程就是对样本空间的每一个样本轮流进行对网络中各层权值的调整学习过程,每个轮回称为一次迭代过程。通过对样本空间的每一个样本进行学习,逐渐减少网络输出的误差,直到达到一定的训练精度。传统的 BP 算法在每次迭代过程中,对样本空间中的样本个体一视同仁都只学习一次,而不管单个样本的个体误差。通过研究 BP 算法的学习样本的过程,我们发现,每个样本的误差是不同的,对那些经过学习后误差比较大的样本个体,说明学习的效

果和强度不够。人脑在记忆知识的时候,也是具有一定选择性,对某些知识记忆比较好,相反对某些知识记忆效果不明显,根据心理学的记忆遗忘规律,对那些记忆效果不好的东西,应该加强记忆的强度和记忆的次数。据此思想,提出对样本空间中的不同个体根据误差的大小,进行不同次数的训练,对那些误差比较大的个体适当地增加学习的次数。

定义样本的期望训练次数为

$$EN_i = 1 + \left[\log_a \frac{E(i) * P}{b \sum_{i=1}^P E(i)} \right] \quad i = 1, 2, \dots, P$$

其中 P 为样本空间中样本的总数, $E(i)$ 为第 i 个样本的误差, a 取大于 1 的数(一般可取 2), b 取 0 到 1 之间的数(一般可取 0.5)。 $[c]$ 表示取非负整运算,即 c 若小于 0,则 c 取 0,否则取下整运算。

这样由上式可见, a 取大于 1 的数,并且取对数运算,这样既保证了每个样本都得到至少学习一次,又仅对那些误差比较大的个体适当地增加学习的次数,防止单个样本训练过度。

2.2 对样本空间中的样本学习顺序的安排

标准 BP 算法是按样本空间中样本的固定顺序进行轮流学习的,基于样本的期望训练数产生训练样本空间,为防止误差比较大的样本紧接着重复训练,对每次产生的样本群,随机打乱样本的顺序,实验表明可以一定程度上避免训练过程中产生的振荡现象,并且加快了收敛的速度。

2.3 自适应改变训练参数 a 和学习率 η

训练参数 a 的大小决定了样本的期望训练次数, a 越大,期望训练次数越小。在训练的早期总误差比较大, a 可以取小一点,在训练的后期,当总误差与要求精度比较接近的时候,可适当增加 a 的大小,减少期望训练次数,防止单个样本训练过度的情况。

学习率 η 也称步长,在标准算法中定为常数,然而在实际应用中,很难确定一个从始至终都合适的最佳学习率。通过研究 BP 网络的误差曲面可以看出,在平坦区域内 η 太小会使训练次数增加,因而希望增大 η 值;而在误差变化剧烈的区域内, η 太大会因调整过大而跨过较窄的“坑凹”出,使训练出现振荡,反而使迭代次数增加。为了加速收敛过程,一个较好的思路是自适应改变学习率,使其该大时增长,该小时减小。改变学习率的办法很多,其目的都是使其在整个训练过程得到合理调节。这里采用下面的方法:

设一初始学习率,经过一批次权值调整后,使总误差增大,则本次调整无效,且 $\eta(t+1) = \beta\eta(t)$, $\beta < 1$;若经过一批次权值调整后,使总误差减小,则本次调整

有效,且 $\eta(t+1) = \theta\eta(t), \theta > 1^{[9-11]}$ 。

2.4 改进的 BP 算法的步骤

- 1)对权值矩阵 W, V 赋随机数,初始化权值矩阵以及其他网络参数。
- 2)随机打乱训练样本的顺序。
- 3)轮流输入训练样本,计算各层输出。
- 4)计算网络的输出误差。
- 5)计算各层的误差信号。
- 6)调整各层的权值。检查是否对所有样本完成一轮训,如完成转步骤 7),否则返回步骤 3)。
- 7)检查原样本空间的网络总误差是否达到精度要求,如达到则停止,否则,根据样本期望训练数产生下一次训练的样本,返回步骤 2)。

3 仿真实例

实例 1:催化剂配方建模。

这里选择文献[1]中的一个实际问题——催化剂配方建模,来对改进后的算法进行比较研究。

输入结点数 $N = 5$,输出结点数 $L = 3$,隐层结点数 $M = 4$,样本数 $P = 18$ 个,对应学习率 $\eta = 0.8$,期望训练参数 $a = 2$,训练精度 $\epsilon = 0.001$,共进行 20 次计算,计算结果如表 1 所示,误差随迭代次数变化如图 2 所示。

表中的平均迭代次数是对两种算法分别以不同的初始权值实验了 20 次的平均值,平均总的迭代时间是平均迭代次数与平均每次迭代时间的乘积。图中,1 为标准 BP 算法,2 为文中 BP 算法。

表 1 实例 1 算法时间复杂度比较

算法	平均迭代次数	平均每次迭代时间/ms	平均总的迭代时间/s
标准 BP 算法	5286	0.19	1.01
文中 BP 算法	1251	0.20	0.25

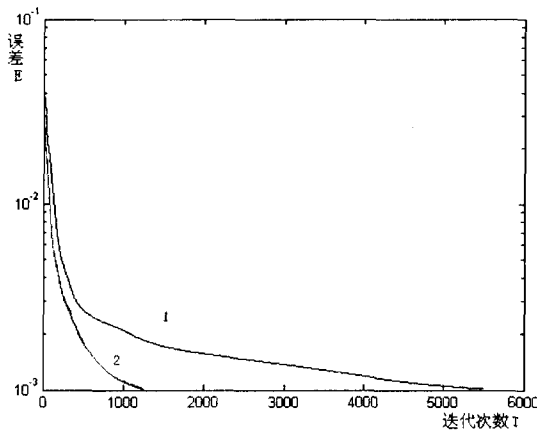


图 2 实例 1 误差变化图

从表 1 和图 2 可以看出,文中 BP 算法所需迭代次数和总的学习时间都有明显减少。虽然文中 BP 算法

的单次迭代工作量比标准 BP 有所增加,这是因为每次迭代训练样本的个数有所增加,但总的耗时减小了 4 倍多。从图 2 的误差变化曲线也可看出学习速率明显提高。

实例 2:parity 问题。

考虑有 3 个输入的 Parity 问题,也即若输入的模式中 1 的个数为奇数,则要求输出为 0、否则为 1,取网络结构为 3-2-1,训练样本共有 8 组。训练精度 $\epsilon = 0.000000001$,共进行 20 次计算,计算结果如表 2 所示,误差变化如图 3 所示。

表 2 实例 2 算法时间复杂度比较

算法	平均迭代次数	平均每次迭代时间/ms	平均总的迭代时间/s
标准 BP 算法	79039	0.015	1.19
文中 BP 算法	16499	0.016	0.26

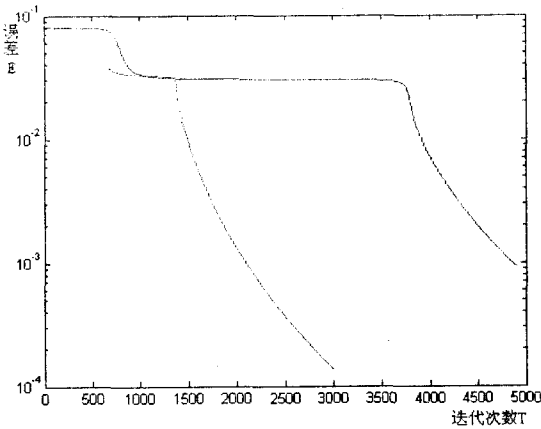


图 3 实例 2 误差变化图

从表 2 可以看出,文中 BP 算法所需迭代次数和总的学习时间比标准 BP 算法减小了近 5 倍。图 3 画出了前 5000 次迭代的误差变化曲线,可以看出文中 BP 算法更能克服局部最优解的情况,学习速率明显提高。

实例 3:函数逼近。

函数: $y = 0.4\sin(2\pi x) + 5$

多层前向网络的一个重要功能就是函数逼近,为了比较文中算法与标准 BP 算法的函数逼近效果,用 1-40-1 三层神经网络来逼近上面的函数,训练样本为在 $[0,1]$ 之间随机取 50 个值作为函数的输入 x 及其对应的 y 值。两种算法分别经过不同的迭代次数达到训练精度 $\epsilon = 0.001$ 。为检验函数逼近效果,以 x 在 $[0,1]$ 上随机取 50 个值作为网络的输入,共进行 20 次计算,计算结果如表 3 所示,误差变化如图 4 所示。

表 3 实例 3 算法时间复杂度比较

算法	平均迭代次数	平均每次迭代时间/ms	平均总迭代时间/s	平均检验精度
标准 BP 算法	1010	0.20	0.20	0.0020
文中 BP 算法	455	0.21	0.09	0.0016

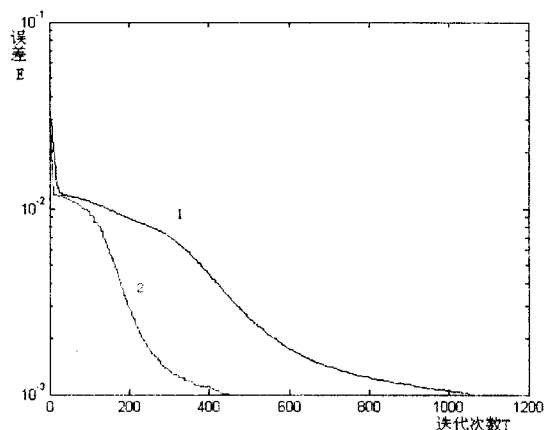


图 4 实例 3 误差变化图

从表 3 可以看出:①文中算法收敛速度快,比标准 BP 算法提高了 2 到 3 倍,这点与前面两个仿真结果是一致的。②另外,文中 BP 算法的检验误差比 BP 算法的平均检验误差也有所减小。图 4 也说明了学习速率明显提高。

4 结束语

从心理学中的记忆遗忘规律出发,提出了一种基于样本期望训练数的改进 BP 算法。通过几个仿真计算的结果,表明文中 BP 算法与标准 BP 算法相比,有收敛速度快、克服局部最优解强、逼近精度高的优点。重要的是文中 BP 算法具有通用性,可以与变步长法、加动量项法、引入陡度因子等其他训练方法相结合使用,以进一步提高算法的收敛速度,在以后将深入进行

这方面的工作。

参考文献:

- [1] Rumelhart D E, Williams R J. Learning internal representations by error-propagation[C]//In Parallel Distributed Processing. Cambridge:MA MIT Press, 1986:318-362.
- [2] Weymacer N, Martens J P. A fast and robust learning algorithm for feedforward neural networks[J]. Neural Networks, 1991, 4(3):363-369.
- [3] 韩力群. 人工神经网络教程[M]. 北京:北京邮电大学出版社, 2006.
- [4] Vapnik V N. The Nature of Statistical of Learning Theory [M]. New York:Spring, 1995.
- [5] 罗莉, 罗强, 胡守仁. 前馈网络的一种优质高效学习算法[J]. 计算机研究与发展, 1997, 34(2):107-112.
- [6] 向国全, 董道珍. BP 模型中的激励函数和改进的网络训练法[J]. 计算机研究与发展, 1997, 34(2):113-117.
- [7] 贾丽会, 张修如. BP 算法改进与分析[J]. 计算机技术与发展, 2006, 16(10):101-103.
- [8] 袁曾任, 沈晓峰. 加快基于 BP 算法神经网络学习的新方法[J]. 清华大学学报:自然科学报, 1995, 35(3):1-6.
- [9] 徐宗本, 张进社, 郑亚林. 计算智能中的仿生学:理论与算法[M]. 北京:科学出版社, 2004.
- [10] 邓志东, 孙增圻. 利用线性再励的自适应变步长快速 BP 算法[J]. 模式识别与人工智能, 1993, 6(4):319-323.
- [11] 雷鸣, 尹申明, 杨叔子. 神经网络自适应学习研究[J]. 系统工程与电子技术, 1994(3):19-27.

(上接第 102 页)

好的应用框架可大大提高软件开发效率和团队协作工作的能力。文中在分析 Web 应用实现方式的基础上, 提出一种新的 Web 应用框架, 其通过构建独立的业务逻辑接口层和数据访问接口层, 实现了表示层、业务功能层、数据存储层之间的松散耦合, 该框架在某大型制造企业的工时管理系统开发中得到了很好的应用。

文中给出的软件框架, 不仅为快速构建结构清晰的 Web 应用提供了一种有效方法, 使开发出的系统具有较强的可移植性和灵活性, 而且可减低团队开发中的信息流量, 有效提高团队协作开发的效率。

参考文献:

- [1] 杨晓江, 王继成. .NET Web 应用的 MVC 设计[J]. 计算机工程与应用, 2003, 39(26):153-155.
- [2] Atkinson C, Bunse C, Hans-Gerhard T. Towards a General Component Model for Web-Based Applications[J]. Annals of

Software Engineering, 2002, 13:47-52.

- [3] 王映辉, 王英杰, 王彦君, 等. 基于 MVC 的软件界面体系结构研究与实现[J]. 计算机应用研究, 2004(8):188-190.
- [4] 吴立春. 基于 Facade 模式的业务逻辑层框架设计[J]. 微计算机信息, 2007(24):245-246.
- [5] Rokitskii R B. A Six-Level Architecture for Development of Internet Applications[J]. Cybernetics and Systems Analysis, 2002, 38(6):921-925.
- [6] 纪颖, 马光思. 使用 DAO 和业务代理联合模式整合 Web 应用框架[J]. 计算机技术与发展, 2006, 16(11):19-22.
- [7] Gun Ho Lee, Junsu Jung. Web framework with Java and XML in multi-tiers for productivity[J]. Future Generation Computer Systems, 2007, 23(2):263-268.
- [8] 庞开放, 李龙澍. 基于 .NET 框架的 Web 应用设计与实现[J]. 微机发展(现更名:计算机技术与发展), 2005, 15(3):85-89.