第 34 卷 第 1 期 计 算 机 工 程 2008 年 1 月
Vol.34 No.1 Computer Engineering January 2008

·人工智能及识别技术· 文章编号：1000—3428(2008)01—0215—03 文献标识码：A 中图分类号：TP301.6

基于 LMBP 改进算法的神经网络结构优化

杨 英[1]，唐 平[2]，王越超[1]，丘衍航[1]

(1. 广东工业大学计算机学院，广州 510006；2. 广东工业大学自动化学院，广州 510006)

摘 要：提出了一种基于增长法的神经网络结构优化算法。在函数逼近的 BP 神经网络中引入一种改进的 BP 算法(LMBP 算法)，通过二次误差下降与梯度下降，利用误差变化规律分析网络结构的优化程度，自适应地增加隐层神经元或网络层次，从而得到一个合适的网络结构。进行了仿真实验及该算法与 RAN 算法用于逼近函数的对比实验，实验结果表明了该算法的有效性。

关键词：神经网络；结构优化；LMBP 算法；函数逼近；RAN 算法

Neural Network Structure Optimization Based on Improved LMBP Algorithm

YANG Ying[1], TANG Ping[2], WANG Yue-chao[1], QIU Yan-hang[1]

(1. Computer Faculty, Guangdong University of Technology, Guangzhou 510006;
2. Automation Faculty, Guangdong University of Technology, Guangzhou 510006)

【Abstract】An algorithm to optimize artificial neural networks structure based on constructive method is presented. And an improved BP algorithm, LMBP algorithm, is introduced about simplest BP neural network for function approximation. By using the rules of error changing based on quadratic error and gradient reducing, this paper analyzes the optimization of the network structure. By adding hidden neurons or network layers adaptively, a proper structure of the network is got. Simulation experiments are provided to compare the approach with RAN algorithm for solving function approximation. The results show the effectiveness of the algorithm.

【Key words】 neural networks; structure optimization; LMBP algorithm; function approximation; RAN algorithm

1 概述

神经网络的理论和应用研究虽已得到极大的发展，而且渗透到几乎所有的工程应用领域，但至今仍没有一个系统有效解决神经网络结构设计的方法。网络结构设计问题之所以重要是因为如果神经网络结点个数过多，则网络结构庞大，网络泛化能力降低，导致过模拟；如果网络结点过少，则不能对所要解决的问题形成一个好的模型，导致不收敛。这方面的研究存在以下难点：(1)可用的信息不足。学习之前，待解决的问题中往往没有足够的信息显示应该使用多大规模的神经网络。(2)隐层单元的知识表述模糊[1]。通常，为了保证测试误差足够小，需要用足够多的隐层单元来达到所要求的精度。学习过的知识往往分布在多个隐层单元上，因此，很难表述单个隐层单元所学的知识。(3)局部最小化问题很难避免。因为传统的方法是学习算法，这个问题会随着网络规模的增大变得越来越严重。解决局部最小化问题，除了优化权值与阈值外，还有必要通过减少不必要的连接与阈值优化神经网络的结构。

对于神经网络结构的优化问题，许多学者提出了创造性的研究成果[2]，如Tiling算法[3]、Upting算法[4]、CC算法[5-6]等。但Tiling算法无法推广到实数映射，Upting算法要求事先知道样本的大小，而CC算法会导致过度拟合。此外还有修剪法[7-8]及基于进化的方法，但前者计算量太大，后者编码抽象、解释困难。在实际应用中，一般采用凑数的方法。仅凭感觉或经验构造网络结构，尚无较好的理论指导，往往不易找到合适的网络结构。

本文提出了一种新的函数逼近的BP网络结构优化方法，其本质上是一种递增方法。虽然与文献[9]提出的资源分配网络(RAN)同为BP算法的改进算法且都先假设一个最简单的网络，但RAN在学习过程中一旦发现“未建模”的样本就分配新的计算节点，这种串行学习有可能为“不好”的样本分配隐节点，而本文提出的算法本质上是并行的批学习算法。本文先引进LMBP算法，然后改进这个算法中的对学习误差的判断，动态地检查网络性能。如有限代数后误差不能达到预定要求，并且权值与阈值不再改变，说明该结构的网络不再满足问题要求，则先增加隐层单元数量，如还不能满足问题要求，再增加网络层次。仿真实验表明，该算法正确有效且稳定性较好，具有推广能力；对比实验也表明经算法优化的网络性能优于RAN解决此类问题的性能。

2 函数逼近的神经网络结构修改原理及其算法

对 3 层BP网，许多人证明了以下的万能逼近定理：含一个隐层的 3 层BP网络，只要隐节点数足够多，就能以任意精度逼近有界区域上的任意连续函数。我国陈天平等指出，神经元函数的有界性(不是连续性)是保证任意逼近的充分条件。尽管万能逼近定理说明了神经网络有一个隐层就能实现任意逼近，但这不表明 3 层的网络结构就是最合理的。事实

基金项目：广东省自然科学(博士启动)基金资助项目“基于可拓推理的全自主足球机器人系统冲突消解的研究”(04300167)

作者简介：杨 英(1978－)，女，硕士研究生，主研方向：智能工程与软计算；唐 平，副教授、博士；王越超、丘衍航，硕士研究生

收稿日期：2007-02-15 **E-mail：**yangyingana@tom.com

上，对于同一目标函数，4 层的BP网有时候可能比 3 层的BP网使用更少的神经元。尽管BP算法是目前神经网络学习中应用最广的学习算法之一，但该算法基于最速下降法，因此，它具有最速下降法的固有缺点：易陷入局部极小，收敛速度慢和引起振荡效应。为了克服以上缺点，本文引进一个改进的BP算法——LMBP算法。在最优化理论中，Levenberg-Marquart(LM)算法[6,10]是一种非常有效的优化设计方法,从收敛速度和收敛性来看，它是牛顿法和最速下降法的一种折中算法。

2.1 LMBP 算法在网络结构优化中的改进

在 LMBP 算法中，学习步长 λ^{-1} 对算法效果起着关键作用，当 λ 增大到一个极限时，学习误差不再变小、权值向量不再改变，继续学习变得没有意义，此时应当调整网络结构。这种误差变化的规律使得 LM 算法可用于神经网络结构的优化中。

本文中，输入向量定为一维$\boldsymbol{x}=[x_1]$，输出向量同为一维，期望输出$t=[t_1]$，实际输出向量为$\boldsymbol{y}=[y_1]$。设目标函数为

$$E=\frac{1}{2}\sum_{p}(\varepsilon^{p})^{2}=\frac{1}{2}\|\varepsilon\|^{2} \tag{1}$$

其中，p 为样本个数；ε 为期望输出与实际输出的间的误差向量；$\boldsymbol{w}^{\text{old}}$ 向 $\boldsymbol{w}^{\text{new}}$ 移动。

$$\boldsymbol{\varepsilon}(\boldsymbol{w}^{\text{new}})=\boldsymbol{\varepsilon}(\boldsymbol{w}^{\text{old}})+\boldsymbol{Z}(\boldsymbol{w}^{\text{new}}-\boldsymbol{w}^{\text{old}})$$

其中，$\boldsymbol{Z}$ 元素为 $(Z)_{pi}=\dfrac{\partial\boldsymbol{\varepsilon}^{p}}{\partial\boldsymbol{w}_i}$。

算法最后得到的权值修改公式[10]为

$$\boldsymbol{w}^{new}=\boldsymbol{w}^{old}-(\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{Z}+\lambda\boldsymbol{I})^{-1}\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{\varepsilon}(\boldsymbol{w}^{old}) \tag{2}$$

最开始定义网络结构为 1-1，权值向量 $\boldsymbol{w}=[w_{11},w_{12}]$。为简化计算，这里把阈值当作权值处理。神经网络的规模体现在权值向量上。

在式(2)的迭代中，步长 λ 在不断变化。这里采用常用的做法，起始为 $\lambda=0.1$。使用式(2)后会出现 2 种情况：(1)误差减小，则保留 $\boldsymbol{w}^{\text{new}}$，$\lambda$ 乘以 1/10，重复此步骤；(2)误差增加，则维持 $\boldsymbol{w}^{\text{old}}$，$\lambda$ 乘以 10，然后按式(2)计算 $\boldsymbol{w}^{\text{new}}$。当第(2)种情况不断发生，$\lambda$ 持续增大，而 $\boldsymbol{w}^{\text{old}}$ 一直没有更新，误差在迭代多次后仍未减小，就需要修改神经网络结构：在隐层单元增加新的神经元。权值向量维数增加的数量取决于此隐层的前一层及后一层神经元的个数。设它的前一层神经元个数为 m，后一层为 l，则增加的权值向量个数为 $m+l+1$ 个(1 个阈值)。因为最开始只定义了 2 层，所以这时要增加一层，网络规模变化为 1-1-1，权值向量变为 $\boldsymbol{w}=[w_{11},w_{12},w_{21},w_{22}]$。实际上，网络的隐层数和隐节点决定了网络的规模，而网络的规模与其性能密切相关。神经网络的规模越大，网络中的自由参数就越多；反之，网络中的自由参数就越少。通过增加隐层神经元扩大规模，随之增加网络中的自由参数，有利于逼近目标函数。之后的学习中会给出一组新的权值向量重新赋值，重新迭代式(2)。若迭代若干次后误差仍达不到预期要求且不再下降，权值也没被修改，则继续通过修改权值向量来改变网络规模。总结而言，设神经网络此时规模为

$$\boldsymbol{w}=[w_{11},w_{12},\cdots,w_{im},w_{j1},\cdots,w_{jn},w_{k1},\cdots,w_{kl},\cdots]$$

其中，$j=i+1$；$k=j+1$，则在第 j 层增加一个神经元后，权值向量变化为

$$\boldsymbol{w}=[w_{11},w_{12},\cdots,w_{im},w_{j1},\cdots,w_{jn},w_{jn+1},\cdots,w_{jn+m+l+1},w_{k1},\cdots,w_{kl},\cdots] \tag{3}$$

这里可以给隐层神经元数量设一个上限，当某隐层神经元个数已增长到上限，则增加一个新的隐层，依然使用式(3)，只须将下标作简单修改即可。当神经网络达到一个合适的规模时，误差迅速下降，将很快达到预期要求，此时神经网络的结构就是合适的结构了。

2.2 基于改进的 LMBP 算法优化网络结构的算法

改进后的 LMBP 算法用于网络结构优化，本文中最多使用 4 层网络。神经网络结构优化算法如下：

(1)给定一个最小的网络结构：1-1，即一个输入一个输出，无隐层。

(2)用改进的 LM 算法训练网络，模拟函数输出，并追踪期望与实际输出之间的均方差。

(3)若均方差达到要求，则算法结束，否则转到(4)。

(4)若网络结构达到上限，转到(5)，否则按式(3)修改结构：

1)若此时是 2 层结构，则增加至 3 层；

2)若网络已达到 3 层网络的上限，则增加至 4 层，否则，在第 1 隐层增加神经元；

3)若此时是 4 层结构，在第 2 隐层增加神经元，其余结构不变；转到(2)。

(5)算法结束，并显示结果。

3 实验仿真与分析

为了检验本文的网络结构优化算法的有效性，在 Matlab 平台上进行了仿真实验。实验中，学习用的样本是随机分布的，而测试所用的样本是均匀分布的，测试样本的期望输出与实际输出的均方差-测试误差代替泛化误差。实验包括 3 方面内容。

实验 1 网络结构优化算法的有效性检验。包括算法的时间、精度、泛化能力 3 方面的输出。实验对象为 Hermit 多项式：$f=1.1*(1-x+2*x^2)*\exp(-x^2/2)$。训练中给定的训练样本与测试样本都为 100 个，算法所要达到的精度小于等于 0.01。

表 1、图 1~图 3 是实验 1 的结果。在图 1 中，“+”曲线代表函数对应样本的期望输出曲线，而光滑的曲线代表的是训练后网络对测试样本的实际输出曲线。图 2 的曲线代表了误差的在网络训练中的变化。可以看出，误差下降得很快，但是到达一定精度后因网络结构的影响，误差不再变化（对应曲线中平滑的那些段），这个时候调整网络的结构，直至误差达到要求。图 3 是测试误差变化曲线。在表 1 中，网络结构 1-5-1 代表 3 层神经网络；第 1 隐层 5 个神经元，无第 2 隐层，其余神经网络结构描述类似。从这些结果来看，此算法正确可行，而且稳定性高。

表 1 改进 LM 算法逼近 Hermit 函数时输出结果

实验次数	训练代数	时间/min	网络训练后结构	训练误差	测试误差
1	477	0.601 8	1-4-1	0.009 9	0.325 9
2	683	0.967 2	1-5-1	0.009 4	0.818 3
3	609	0.869 0	1-5-1	0.008 8	0.269 7

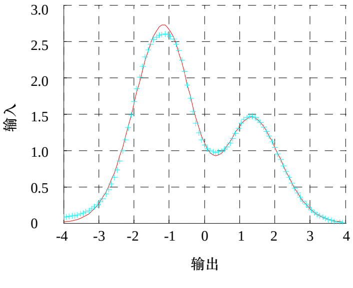

图 1 学习完成后的目标曲线和神经网络输出曲线

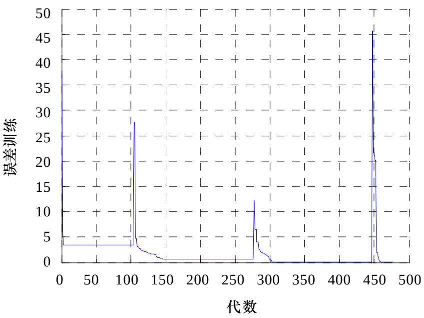

图 2　学习过程中训练误差变化曲线

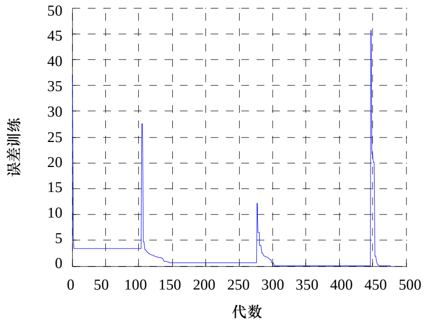

图 3　学习过程中测试误差变化曲线

实验 2 与 3 层 RAN 网络的处理 Hermit 多项式逼近问题的比较。在 Matlab 仿真平台上输出 3 层 RAN 网络处理 Hermit 多项式逼近问题的曲线，图 4~图 6 是 RAN 算法训练的结果。RAN 算法中训练样本 400 个，测试样本 101 个。学习后，隐节点数为 31，测试误差为 1.548 8，总共训练 400 代。

对比实验 1 与实验 2 的结果可以看出，3 层 RAN 模拟同样的函数所需隐节点数(31)比基于改进的 LMBP 的神经网络结构学习算法所需的隐节点数(4 个)，数量上多了许多，网络规模随之大了许多，而代表泛化能力的测试误差为 1.548 8，也较实验 1 中 3 次实验的平均测试误差(0.471 3)大了许多。

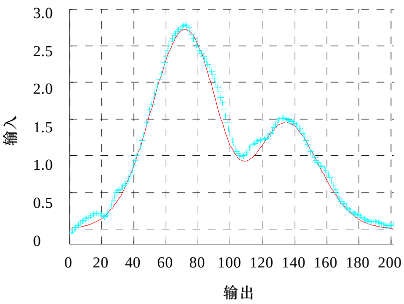

图 4　学习完成后的目标曲线和神经网络输出曲线

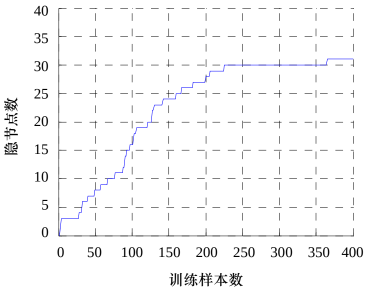

图 5　学习过程中隐节点数的变化

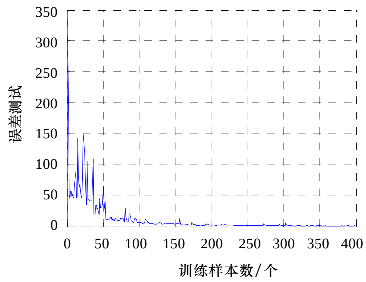

图 6　学习过程中测试误差变化曲线

实验 3 网络结构优化算法的推广能力测试。本文选用 3 个不同的函数进行模拟实验。表 2 的训练结果表明本算法对不同函数均能训练出适合的网络结构。

表 2　改进的 LMBP 算法对不同函数训练后的结果

次数	模拟函数	网络训练后的结构	训练误差	测试误差
1	$f=x^3$	1-3-1	0.006 1	0.066 2
2	$f=(\sin(x)\cdot\sin(x)-0.5)/(1+x\cdot x)$	1-5-1	0.009 9	0.265 7
3	$f=x\cdot\sin(10x)\cdot\cos(x)$	1-5-5-1	0.050 0	2.926 4

4　结束语

本文提出一种优化神经网络结构的方法，利用理论较成熟的 BP 网络，从函数逼近的问题出发，利用 LMBP 算法及误差变化规律改进算法，从而动态优化网络结构。实验表明本算法的有效性、稳定性及一定的推广能力。经算法训练的网络结构精简、泛化能力好、收敛性强。研究还发现，用改进的 LMBP 算法，不同的函数会优化出不同规模网络结构，下一步的工作是将研究函数的复杂程度对网络结构的影响。

参考文献

[1] Takahama T, Sakai S. Structural Learning of Neural Networks by Coevolutionary Genetic Algorithm with Degeneration[C]//Proc. of IEEE Int'l Conference on Systems, Man and Cybernetics. [S. l.]: IEEE Press, 2004.

[2] Widrow B, Hoff M. Adaptive Switching Circuits[M]//IRE Wescon Convention Record. New York: Institute of Radio Engineers, 1960.

[3] Hopfield J. Neural Networks and Physical Systems with Emergent Collective Computational Abilities[J]. Proceedings of the National Academy of Science, 1982, 79(2): 2554-2558.

[4] Rumelhart D E, Hinton G E, Williams R J. Learning Representations by Back-propagation Errors[J]. Nature, 1986, 323(6088): 533-536.

[5] Haykin S. Neural Networks: A Comprehensive Foundation[M]. 2 版. 北京: 清华大学出版社, 2001.

[6] Hagan M. Neural Network Design[M]. 北京: 机械工业出版社, 2002.

[7] 张鸿宾. 训练多层网络的样本数问题[J]. 自动化学报，1993, 19(1): 71-77.

[8] 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.

[9] Platt J. A Resource-allocating Network for Function Interpolation[J]. Neural Computation, 1991, 3(2): 213-225.

[10] 阎平凡, 张长水. 人工神经网络与模拟进化计算[M]. 2 版. 北京: 清华大学出版社, 2005.

[11] 魏海坤. 神经网络结构设计的理论与方法[M]. 北京: 国防工业出版社, 2005.

[12] 余永权. 模糊逻辑控制与神经网络[M]. 北京: 电子工业出版社, 1999.