第45卷 第7期 哈尔滨工业大学学报 Vol. 45 No. 7
2013年7月 JOURNAL OF HARBIN INSTITUTE OF TECHNOLOGY July 2013

填充函数法改进的BP网络SPDS算法

张少仲[1]，李龙锁[2]，任世军[2]，蒋　波[1]，白　英[1]，张维石[1]

（1.大连海事大学 信息学院，116026 大连；2.哈尔滨工业大学 计算机科学与技术学院，150001 哈尔滨）

摘　要：BP网络的训练算法的一个普遍问题是易陷入局部极小.为了解决SPDS算法的这个问题,针对其特点设计了一组新填充函数.通过对该填充函数的分析,证明了用它代替目标函数进行搜索的等价性,并据此改进了SPDS算法.算法的仿真试验证明：当SPDS算法陷入局部极小点时,用设计的填充函数代替目标函数,从而使算法不受局部极小问题的羁绊,可以快速收敛到全局极小点.

关键词：BP网络;填充函数法;局部极小问题

中图分类号：TP39　　**文献标志码**：A　　**文章编号**：0367-6234(2013)07-0126-03

Improved SPDS algorithm in BP network based on filled function method

ZHANG Shaozhong[1], LI Longsuo[2], REN Shijun[2], JIANG Bo[1], BAI Ying[1], ZHANG Weishi[1]

(1. Information School, Dalian Maritime University, 116026 Dalian, China;
2. School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

Abstract: To solve the problem that the BP network training algorithm is easy to fall into local minimal point, this paper designs a set of new filled function in view of its characteristic, which can be used to replace the objective function for search and improve the SPDS algorithm. The algorithm simulation test proves that, when the SPDS algorithm falls into local minimum point, the use of the filled function instead of objective function makes the algorithm avoid the fetters of minimum problem, and accelerates convergence to the global minimum point.

Key words: BP network; Filled function method; Local minimum problem

BP算法的主要缺点之一是训练容易陷入局部极小点[1],而解决局部极小问题的全局优化算法又分为:1)随机型优化算法,如模拟退火技术等[2];2)确定型优化算法,如填充函数法等[3].从减少迭代次数的角度来看,如果能使用确定型优化算法解决问题是最理想的.SPDS算法[4-5]也存在局部极小值问题.为了解决它的局部极小问题,本文针对SPDS算法的特点设计了一组填充函数,而当算法收敛到某局部极小点时,用该点的填充函数来代替目标函数进行搜索.由于用填充函数搜索可以跳出此极小点,并收敛到更低的极小点(如果有的话),又由于本文证明了填充函数的极小点既是目标函数的极小点,所以这种方法不仅摆脱了局部极小点而且还将收敛到另一更低的极小点,从而提高了收敛速度.本文只研究3层BP网络的误差函数为

$$E = \sum_p \sum_i (O_{p_i} - y_{p_i})^2.$$

式中:O_{p_i} 为输出层第 i 个节点对应于第 p 个训练样本的实际输出; y_{p_i} 为相应的期望输出值.

$$E = \sum_p \sum_i \left(g\left(\sum_j w_{ij}\bar{b}_j + \theta_i\right) - y_{p_i}\right)^2 = \sum_p \sum_i \left(g\left(\sum_j w_{ij} g\left(\sum_k w_{kj} b_k + \varphi_j\right) + \theta_i\right) - y_{p_i}\right)^2.$$

式中: w_{ij} 为第 i 个输出层单元到第 j 个隐层单元的权值; w_{kj} 为第 k 个输入层单元到第 j 个隐层单元的权值. $\bar{b}_j$ 为第 j 个隐层单元的输出; b_k 为第 k 个输入层单元的输出. θ_i 为第 i 个输出层单元阈值; φ_j 为第 j 个隐层单元阈值. g 为Sigmoid函数 $1/(1+\exp(-\lambda x))$; $g' = \lambda g(1-g)$.

收稿日期: 2012-10-15.
基金项目: 国家自然科学基金资助项目(61173034).
作者简介: 张少仲(1965—),男,高级工程师.
通信作者: 张少仲, zhang_shaozhong2001@yahoo.com.cn.

神经网络的训练就是使 $\theta_i, w_{ij}, \varphi_j, w_{kj}$ 这 4 组变量取适当的值,使得误差函数 E 取到全局极小值.

1　SPDS 算法

基于单参数动态搜索算法的 SPDS 算法,就是将这 4 组变量逐一作为变量处理,其他值均用上次迭代后的储存值代替. 这样 4 个误差函数分别记为 E_1、E_2、E_3 和 E_4,它们都是关于变量的连续可微函数. 不难看出,每个误差函数均为以下形式:$E_i = \Sigma f_i^2(x)$,其中:$f_i^2(x)$ 为一元函数,$i = 1,2,3,4$.

本文仅考虑针对 θ_i(i 取某个 i_0 时) 为变量的情况,即目标函数为

$$E_1(x) = \sum_p \left(g\left(\sum_j w_{i_0j}\bar{b}_j + x\right) - y_{p_{i_0}}\right)^2 = \sum_p (g(A + x) - B)^2. \tag{1}$$

其中:$A = \sum_j w_{i_0j}\bar{b}_j$; $B = y_{p_{i_0}}$.

2　填充函数法

填充函数法最早由葛人溥提出,它的基本思想是通过构造一个辅助函数,从目标函数的当前局部极小点找到另一个使目标函数值更小的极小点,从而跳出局部极小点.

针对文献[3]中提出的函数中的参数不易调节、假设过多、实际问题不易满足等问题,随后的一些学者进行了改进[6-10],但是基本思想没有变.

本文也借助这个基本思想,针对 SPDS 算法的特点设计填充函数,解决局部极小问题. 考虑优化问题(P):$\min f(x)$,其中:$x \in R^n$,f:$R^n \to R$ 是可微函数. 假设 x^* 是问题(P) 的当前极小点.

定义 1　函数 $p(x,x^*)$ 称为 $f(x)$ 在局部极小点 x^* 处的填充函数,如果 $p(x,x^*)$ 满足条件:

1) 对任意的 $x \in \Omega_1$,有 x^* 是 $p(x,x^*)$ 的严格局部极大点而且 $f(x)$ 与 $p(x,x^*)$ 的单调性一致,这里 $\Omega_1 = \{x \in R^n \mid f(x) < f(x^*)\}$.

2) 对任意的 $x \in \Omega_2$,$p(x,x^*)$ 的梯度不为零,即没有极值点或鞍点,这里 $\Omega_2 = \{x \in R^n \mid f(x) \geqslant f(x^*), x \neq x^*\}$.

条件 1) 保证了在比局部极小点的函数值还小的定义域区间内,填充函数的极小点即为目标函数的极小点;条件 2) 保证了在比局部极小点的函数值还大的定义域区间内,填充函数没有极值点. 因此,当搜索到某局部极小点时,用填充函数代替目标函数继续搜索,得到的极小点值将比该局部极小点的函数值更小,从而跳出局部极小点.

对于式(1) 中目标函数 $E_1(x)$,构造如下的辅助函数为

$$p(x,x^*) = \begin{cases} E_1(x) - E_1(x^*), & x \in \Omega_1 = \{x \in R^n \mid E_1(x) < E_1(x^*)\}, \\ E_1(x) + \overline{E_1}(x), & x \in \Omega_2 = \{x \in R^n \mid E_1(x) \geqslant E_1(x^*), x \neq x^*\}. \end{cases} \tag{2}$$

其中:$\overline{E_1}(x) = \sum_p \left(g\left(\sum_j w_{i_0j}\bar{b}_j + x\right) + y_{p_{i_0}}\right)^2 = \sum_p (g(A + x) + B)^2$.

下面证明该 $p(x,x^*)$ 是一个满足定义 1 的填充函数.

定理 1　式(2) 所定义的 $p(x,x^*)$ 是一个满足定义 1 的填充函数.

证明　对任意的 $x \in \Omega_1$,因为 $E_1(x) < E_1(x^*)$,而 $p(x,x^*) = E_1(x) - E_1(x^*) < 0 = E_1(x^*) - E_1(x^*) = p(x^*,x^*)$,即此时 x^* 是 $p(x,x^*)$ 的严格局部极大点. 又因为 $p'(x,x^*) = E'_1(x)$,故两者的单调性一致,定义 1 的条件 1) 满足.

对任意的 $x \in \Omega_2$,$p(x,x^*) = E_1(x) + \overline{E_1}(x)$,故 $p'(x,x^*) = E_1'(x) + \overline{E_1'}(x)$.

而 $E_1(x) = \sum_p (g(A + x) - B)^2$,注意到 $g' = \lambda g(1 - g)$.

所以 $E_1'(x) = \sum_p 2\lambda(g(A + x) - B)g(A + x)(1 - g(A + x))$,同理可得

$$\overline{E_1'}(x) = \sum_p 2\lambda(g(A+x) + B)g(A+x)(1 - g(A+x)).$$

故

$$\begin{aligned} p'(x,x^*) &= E_1'(x) + \overline{E_1'}(x) = \\ &\sum_p 2\lambda(g(A+x) - B)g(A+x)(1 - g(A+x)) + \\ &\sum_p 2\lambda(g(A+x) + B)\,g(A+x)\,(1 - g(A+x)) = \\ &\sum_p 4\lambda g(A + x)^2(1 - g(A + x)) > 0. \end{aligned} \tag{3}$$

定义 1 的条件 2) 满足. 因此式(2) 所定义的 $p(x,x^*)$ 是一个满足定义 1 的填充函数.

上述定义的填充函数,没有需要调节的参数,没有过多的假设条件,而且针对 SPDS 算法量身定做,实用性强.

3　改进算法及仿真算例

根据上述理论,提出以下改进算法:

步骤 1　以 x_1 为初始点,应用 SPDS 算法得到目标函数 $E_i(x)$ 的某个极小点 x^*.

步骤 2　如果 x^* 点的目标函数值小于某很小正数，则可以认为该点是全局极小点，迭代停止. 否则认为该点是局部极小点，转到步骤 3.

步骤 3　构造关于 x^* 的填充函数 $p(x,x^*)$.

步骤 4　搜索 $p(x,x^*)$ 的非 x^* 极小点，若得到则令该点为 x_1，转到步骤 1. 否则转到步骤 5.

步骤 5　若迭代次数超出限制，迭代停止. 否则返回步骤 4.

为了证明算法的有效性，进行仿真实验.

仿真算例以 Xor 问题为例.

网络结构为 2 - 2 - 1，初值选取如表 1 所示. 在将第 1 个输入层单元与第 1 个隐层单元的权值作为变量进行搜索时，目标函数与该权值的关系如图 1 所示，从图 1 中可知，在 -4.5 处有一个局部极小点，在 0.5 处有一个全局极小点. 当迭代到 -4.5 附近陷入局部极小点时，在 -4.5 处设计填充函数，图 2 表示该填充函数与该权值的关系.

表 1　Xor 问题的初始值

单元	权值		阈值
	输入层第 1 单元	输入层第 2 单元	
隐层第 1 单元	-4.187 455	-0.967 554	1.052 309
隐层第 2 单元	5.002 114	1.304 661	-1.254 606
单元	权值		阈值
	隐层第 1 单元	隐层第 2 单元	
输出层单元	-3.967 442	4.922 374	-0.569 901

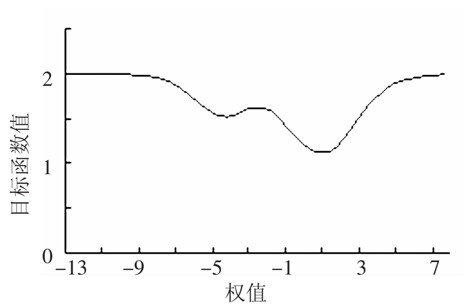

图 1　目标函数值与该权值的关系

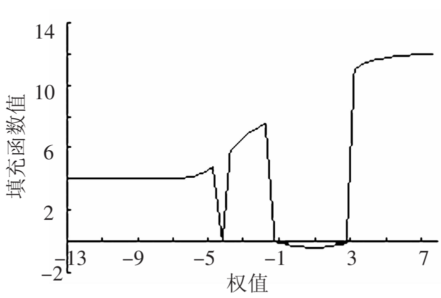

图 2　填充函数值与该权值的关系

从图 2 中可知，在局部极小点 -4.5 周围进行搜索，都不会“陷入”某个“盆域”，使迭代无法跳出该极小点周围. 亦即用填充函数进行搜索可以顺利逃出该局部极小点. 同时全局极小点 0.5 附近是一个比局部极小点 -4.5 还要“低”的“盆域”，可以预见，迭代将在此“盆域”内搜索到比局部极小点 -4.5 还要“低”的极小点 0.5.

由此不难看出，用填充函数思想对 SPDS 算法的改进，可以顺利摆脱局部极小问题的困扰，从而大大增加了快速收敛到全局极小点的可能性.

4　结　论

1）设计了基于填充函数思想的辅助函数，在极小点处用以代替目标函数进行搜索，进而摆脱了局部极小点并可收敛到更“低”的极小点. 从而快速收敛于全局极小点的可能性大大增加了.

2）填充函数的设计简洁明了，有很强的针对性和实用性，并证明了用以代替目标函数进行搜索的等价性.

3）仿真实验证明这种改进是可行的.

参考文献

[1] 张铃，张钹. 神经网络中 BP 算法的分析[J]. 模式识别与人工智能，1994, 7 (3): 191 - 195.

[2] 康立山，谢云，尤矢勇，等. 非数值并行算法（第一册）：模拟退火算法[M]. 北京：科学出版社，1994.

[3] GE R P. Theory of Filled Function Methods For Finding Global Minimizers of Nonlinearly Constrained Minimization Problems [J]. Journal of Computational Mathematics, 1987, 5 (1): 1 - 9.

[4] WANG X F, FENG Y J, ZHAO X. A single parameter dynamic searching algorithm for multi-layer neural networks (Part Ⅰ) [J]. International Journal of Innovative Computing, Information and Control, 2008, 4 (9): 2221 - 2233.

[5] WANG X F, ZHAO X, FENG Y J. A single parameter dynamic searching algorithm for multi-layer neural networks (Part Ⅱ) [J]. International Journal of Innovative Computing, Information and Control, 2008, 4 (11): 2941 - 2954.

[6] 朱文兴，傅清祥. 一个基于填充函数变换的对称 TSP 问题的局部搜索算法[J]. 计算机学报，2002, 25 (7): 701 - 707.

[7] 钟以维，徐应涛，张莹. 用填充函数法改进的人脸比对算法[J]. 计算机技术与发展，2009, 19 (8): 78 - 81.

[8] 王晓丽，周国标. 实现快速全局优化的跨越函数方法[J]. 应用数学，2006, 19 (1): 56 - 60.

[9] 王晓丽，周国标. 跨越函数法：一类全局优化的新策略[J]. 上海交通大学学报，2006, 40 (9): 21 - 24.

[10] 刘津，叶仲泉. 一类新的寻求全局最优解的填充函数[J]. 计算机技术与发展，2010, 20 (6): 36 - 38.

（编辑　张　红）