文章编号:1673-5005(2015)02-00164-07 doi:10.3969/j.issn.1673-5005.2015.02.026

带光滑 $L_{1/2}$ 正则化项的神经网络逆向迭代算法收敛性分析

黄炳家[1], 王 健[1,2], 温艳青[1], 杨喜峰[1], 邵红梅[1], 王 兢[2]

(1. 中国石油大学理学院,山东青岛 266580; 2. 大连理工大学电信学部,辽宁大连 116024)

摘要:$L_{1/2}$正则子比 L_2 正则子更具稀疏性,有更强的剪枝能力;但其非凸、非光滑以及不满足 Lipschitz 条件的函数性质,使神经网络训练过程易于出现数值振荡现象,并且给收敛性分析带来理论困难。用光滑函数逼近 $L_{1/2}$正则子在克服数值振荡的同时可以保证目标函数具有良好的连续可微性质。针对提出的带光滑 $L_{1/2}$正则化项的逆向迭代神经网络模型,证明了误差函数的单调递减性质及算法的确定型收敛性:弱收敛和强收敛。数值实验表明,新的逆向迭代学习算法较已有算法保证了输入向量序列在训练过程中的稳定性及稀疏性,并有较好的泛化能力。

关键词:神经网络; 梯度法; 逆向迭代算法; 单调性; 正则化; 收敛性

中图分类号:TP 183 **文献标志码**:A

引用格式:黄炳家,王健,温艳青,等. 带光滑 $L_{1/2}$正则化项的神经网络逆向迭代算法收敛性分析[J]. 中国石油大学学报(自然科学版),2015,39(2):164-170.

HUANG Bingjia, WANG Jian, WEN Yanqing, et al. Convergence analysis of inverse iterative algorithms for neural networks with $L_{1/2}$ penalty[J]. Journal of China University of Petroleum(Edition of Natural Science), 2015,39(2):164-170.

Convergence analysis of inverse iterative algorithms for neural networks with $L_{1/2}$ penalty

HUANG Bingjia[1], WANG Jian[1,2], WEN Yanqing[1], YANG Xifeng[1], SHAO Hongmei[1], WANG Jing[2]

(1. *College of Science in China University of Petroleum*, *Qingdao* 266580, *China*)

(2. *Electronic Information and Electrical Engineering in Dalian University of Technology*, *Dalian* 116024)

Abstract: Compared with the common existing $L_{1/2}$ penalty term for trained neural networks, the algorithm of neural networks with $L_{1/2}$ penalty shows more sparse performance and prunes neurons more effectively. However, the $L_{1/2}$ penalty is non-convex, non-smooth and non-Lipschitz continuous, which inevitably leads to numerical oscillations and problems in theoretical convergence analysis. It is a better solution by using smooth function to approach the penalty. For the proposed algorithm, the error function decreases monotonously with fixed trained weights. In addition, the weak and strong convergence were proved. The presented algorithm performs more stable and sparse than the existing inverse iterative neural networks, and is applicable to more general cases.

Keywords: neural networks; gradient descent; inverse iterative algorithm; monotonicity; regularization; convergence

反向传播网络模型(BP)是应用最为广泛的神经网络模型,已成功用于股市预测、专家系统、模式识别等领域[1-2]。该神经网络有 3 个本质的缺点:收敛速度慢、网络容错性差以及易陷入局部极小而得不到全局最优解。针对其缺陷,已出现很多高效的改进算法,包括增加惩罚项、自适应调节学习率、引入陡度因子等方法[3-7]。其中,在误差函数中增加正则项是一种典型的解决方案,可以有效提高网络的

收稿日期:2014-12-15

基金项目:国家自然科学基金青年项目(61305075);教育部高等学校博士学科点专项科研基金(20130133120014);中国博士后科学基金面上项目(2012M520624);山东省自然科学基金青年项目(ZR2013FQ004);中央高校基本科研业务费专项基金(14CX05042A,14CX02024A)

作者简介:黄炳家(1964-), 男, 教授, 硕士, 从事智能信息处理方面的研究。E-mail: hbjia@upc.edu.cn。

泛化能力并改善网络的剪枝效果。神经网络逆向迭代算法与反问题有密切的联系。目前,反问题研究已经遍及现代化生产研究的各个领域[8-9]。正如正问题与反问题:传统的 BP 算法在权值空间搜索,由因求果;神经网络逆向迭代算法在输入空间内使用梯度法搜索,由果推因。神经网络逆向迭代算法在工程实际中有广泛的应用价值,Kindermann[10] 首次提出了基于梯度下降法的神经网络逆向迭代学习算法。为解决电磁机构最优化问题,Fanni 等[11-12] 设计出一种新的逆向神经网络模型,有效避免了网络训练过程中易于陷入局部极小值的问题。利用反延迟函数模型,Hayakawa 等[13] 提出了一种类似于 BVP 模型的逆向神经网络,该网络可以快速收敛到待求解的组合优化问题的最优解。基于实时逆向神经网络设计的可重构电子元件可以明显提高声纳系统的输出性能,比已有的粒子群算法具有更大的优势[14]。但是,关于神经网络逆向迭代算法的理论分析却不多见,孟少奇[15] 提出了一种带动量项的神经网络逆向迭代算法并给出了收敛性分析,这一算法有助于改进网络收敛速度,但并不具有稀疏性。在神经网络变量选择问题中,稀疏性是一个重要指标。徐宗本等[16] 提出了基于非凸罚的 $L_{1/2}$ 正则子的重赋权迭代算法,比 L_2 正则子具有更强的稀疏性。把 $L_{1/2}$ 正则子引入神经网络逆向迭代算法可以增强网络泛化能力同时也保证稀疏性。为克服网络训练过程中出现的数值振荡现象以及惩罚项函数非凸、非光滑并且不满足非扩张等性质带来的理论难度,笔者提出带光滑 $L_{1/2}$ 正则子的神经网络逆向迭代算法,并证明算法的确定型收敛性。

1　神经网络逆向迭代算法

1.1　$L_{1/2}$ 正则子

变量选择和特征提取是高维与海量数据处理面临的基本问题。近年来发展起来的正则化方法为求解上述问题提供了一条途径。当一个表示模型中含有冗余变量时,变量选择和特征提取要求在辨识出真实变量的同时剔除其冗余变量。当模型中含有大量的冗余变量时,对应的问题称为稀疏问题。L_0 正则子是最早应用于变量选择和特征提取的正则化方法,它以参数个数为约束给出最优的变量选择结果,通常能产生最稀疏的解,但需要求解一个困难的组合优化问题。L_1 正则化能产生较稀疏的解并且仅需求一个凸优化问题,并不能总是产生最稀疏的解,且对于含有重尾分布的误差数据并不能总是取得好的效果。L_2 正则化产生光滑解,不具有稀疏性。近年来,关于 $L_{1/2}$ 正则子的研究[16] 是统计与机器学习领域所关注的焦点之一,在稀疏意义下它可以替代 $L_p(0<p<1)$ 正则子。

在逆向神经网络训练过程中,加入正则化惩罚项是一种常见的提高网络泛化能力的方法,L_2 正则子是最常用的策略,但是其网络剪枝能力较弱,因此一个自然的想法是在网络训练过程中采用 $L_{1/2}$ 正则话惩罚项以达到改进泛化能力的同时提高网络的剪枝效果。

1.2　$L_{1/2}$ 正则化逆向迭代学习算法

本文中主要考虑一个三层的神经网络,网络中输入层, 隐层和输出层单元个数分别为 p、n、1。输入样本为 $x\in\mathbf{R}^p$,相应的理想输出为 $O\in\mathbf{R}$。连接输入层和隐层的权值矩阵为 $\boldsymbol{V}=(v_{ij})_{n\times p}$,记 $\boldsymbol{v}_i=(v_{i1},\cdots,v_{ip})^{\mathrm{T}}\in\mathbf{R}^p, i=1,2,\cdots,n$,连接隐层和输出层的权向量为 $\boldsymbol{w}=(w_1,w_2,\cdots,w_n)^{\mathrm{T}}\in\mathbf{R}^n$。

记 $g:\mathbf{R}\to\mathbf{R}$ 为隐层和输出层的一个给定的激活函数。为了简化表达式,对 $\boldsymbol{u}=(u_1,u_2,\cdots,u_n)^{\mathrm{T}}\in\mathbf{R}^n$,引入向量值函数

$$G(\boldsymbol{u})=(g(u_1),g(u_2),\cdots,g(u_n))^{\mathrm{T}}. \tag{1}$$

对任意输入 $\boldsymbol{x}\in\mathbf{R}^p$,网络的最终输出为

$$y=g(\boldsymbol{w}\cdot G(\boldsymbol{V}\boldsymbol{x})). \tag{2}$$

神经网络带 $L_{1/2}$ 正则化惩罚项的误差函数为

$$E(\boldsymbol{x})=\frac{1}{2}(O-g(\boldsymbol{w}\cdot G(\boldsymbol{V}\boldsymbol{x})))^2+\lambda\|\boldsymbol{x}\|_{1/2}^{1/2}. \tag{3}$$

其中 $\|\boldsymbol{x}\|_{1/2}^{1/2}=\sum_{j=1}^{p}|x_j|^{1/2}$,$\lambda>0$ 是处罚项系数。

迭代算法的目的是,对给定的输出 $O\in\mathbf{R}$,确定 $\boldsymbol{x}$ 使得误差函数 $E(\boldsymbol{x})$ 达到极小。为简化书写形式,做如下变换:

$$\tilde{g}(t)=\frac{1}{2}(O-g(t))^2,\ t\in\mathbf{R}. \tag{4}$$

那么,误差函数关于输入向量的梯度为

$$E_x(\boldsymbol{x})=\tilde{g}'(\boldsymbol{w}\cdot G(\boldsymbol{V}\boldsymbol{x}))\sum_{i=1}^{n}w_ig'(\boldsymbol{v}_i\cdot\boldsymbol{x})\boldsymbol{v}_i+\lambda\nabla(\|\boldsymbol{x}\|_{1/2}^{1/2}). \tag{5}$$

其中

$$\nabla(\|\boldsymbol{x}\|_{1/2}^{1/2})=\left(\frac{\mathrm{sgn}(x_1)}{2|x_1|^{1/2}},\frac{\mathrm{sgn}(x_2)}{2|x_2|^{1/2}},\cdots,\frac{\mathrm{sgn}(x_p)}{2|x_p|^{1/2}}\right)^{\mathrm{T}},$$

这里 $\mathrm{sgn}(\cdot)$ 为符号函数。

给定初始输入向量 $\boldsymbol{x}^0\in\mathbf{R}^p$,利用最速下降法给出神经网络的逆向迭代学习算法,更新规则如下:

$$\boldsymbol{x}^{k+1}=\boldsymbol{x}^k-\eta E_x(\boldsymbol{x}^k), \tag{6}$$

$$\boldsymbol{x}^{k+1}=\boldsymbol{x}^{k}-\eta\left[\tilde{g}'(\boldsymbol{w}\cdot G(\boldsymbol{V}\boldsymbol{x}^{k}))\sum_{i=1}^{n}w_i g'(\boldsymbol{v}_i\cdot\boldsymbol{x}^{k})\boldsymbol{v}_i+\lambda\nabla(\|\boldsymbol{x}^{k}\|_{1/2}^{1/2})\right]. \tag{7}$$

其中 $\eta>0$ 是学习率。

1.3 光滑 $L_{1/2}$ 正则化逆向迭代学习算法

对于上述形式的 $L_{1/2}$ 正则化逆向迭代算法，一个直观的问题是输入向量序列在训练过程中容易出现数值振荡现象。为克服这一困难，并且从易于理论推导的角度出发，考虑用光滑函数逼近上述算法中的 $L_{1/2}$ 正则化项。误差函数如下：

$$E(\boldsymbol{x})=\frac{1}{2}(O-g(\boldsymbol{w}\cdot G(\boldsymbol{V}\boldsymbol{x})))^2+\lambda\|f(\boldsymbol{x})\|_{1/2}^{1/2}. \tag{8}$$

其中 $\boldsymbol{x}=(x_1,x_2,\cdots,x_p)^{\mathrm{T}}\in\mathbf{R}^p$，向量值函数 $f(\boldsymbol{x})=(f(x_1),f(x_2),\cdots,f(x_p))^{\mathrm{T}}$，$\|f(\boldsymbol{x})\|_{1/2}^{1/2}=\sum_{j=1}^{p}\sqrt{f(x_j)}$，令

$$f(t)=\begin{cases}-t, & t\leqslant -a,\\ -\dfrac{1}{8a^3}t^4+\dfrac{3}{4a}t^2+\dfrac{3}{8}a, & -a<t<a,\\ t, & t\geqslant a.\end{cases} \tag{9}$$

其中 a 是一个小的正常数(图 1)。

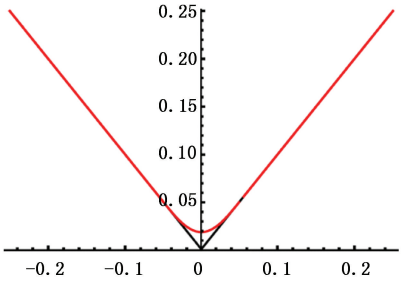

图 1 光滑函数逼近绝对值函数

Fig. 1 Smooth approximation of absolute value

容易知道，其一阶导数和二阶导数计算如下：

$$f'(t)=\begin{cases}-1, & t\leqslant -a,\\ -\dfrac{1}{2a^3}t^3+\dfrac{3}{2a}t, & -a<t<a,\\ 1, & t\geqslant a.\end{cases} \tag{10}$$

$$f''(t)=\begin{cases}0, & t\leqslant -a,\\ -\dfrac{3}{2a^3}t^2+\dfrac{3}{2a}, & -a<t<a,\\ 0, & t\geqslant a.\end{cases} \tag{11}$$

并可得到

$$f(t)\in\left[\frac{3}{8}a,+\infty\right),\ f'(t)\in[-1,1],\ f''(t)\in\left[0,\frac{3}{2a}\right]. \tag{12}$$

那么，误差函数关于输入向量的梯度为

$$E_x(\boldsymbol{x})=\tilde{g}'(\boldsymbol{w}\cdot G(\boldsymbol{V}\boldsymbol{x}))\sum_{i=1}^{n}w_i g'(\boldsymbol{v}_i\cdot\boldsymbol{x})\boldsymbol{v}_i+\lambda\nabla(\|f(\boldsymbol{x})\|_{1/2}^{1/2}), \tag{13}$$

其中

$$\nabla(\|f(\boldsymbol{x})\|_{1/2}^{1/2})=\left(\frac{f'(x_1)}{2f^{1/2}(x_1)},\frac{f'(x_2)}{2f^{1/2}(x_2)},\cdots,\frac{f'(x_p)}{2f^{1/2}(x_p)}\right)^{\mathrm{T}}.$$

给定初始输入向量 $\boldsymbol{x}^0\in\mathbf{R}^p$，基于光滑 $L_{1/2}$ 正则化逆向迭代算法如下：

$$\boldsymbol{x}^{k+1}=\boldsymbol{x}^{k}-\eta E_x(\boldsymbol{x}^{k})=\boldsymbol{x}^{k}-\eta\left[\tilde{g}'(\boldsymbol{w}\cdot G(\boldsymbol{V}\boldsymbol{x}^{k}))\sum_{i=1}^{n}w_i g'(\boldsymbol{v}_i\cdot\boldsymbol{x}^{k})\boldsymbol{v}_i+\lambda\nabla(\|f(\boldsymbol{x}^{k})\|_{1/2}^{1/2})\right]. \tag{14}$$

为表述方便，引入如下记号：

$$\Delta\boldsymbol{x}^{k}=\boldsymbol{x}^{k+1}-\boldsymbol{x}^{k}=-\eta E_x(\boldsymbol{x}^{k}); \tag{15}$$

$$\boldsymbol{G}^{k}=G(\boldsymbol{V}\boldsymbol{x}^{k}); \tag{16}$$

$$\boldsymbol{\psi}^{k}=\boldsymbol{G}^{k+1}-\boldsymbol{G}^{k}. \tag{17}$$

2 主要结果和收敛性证明

对任意的 $\boldsymbol{x}\in\mathbf{R}^p$，记 $\|\boldsymbol{x}\|=\sqrt{\sum_{j=1}^{p}(x_j)^2}$，其中 $\|\cdot\|$ 为欧氏范数。设 $\Omega\subset\mathbf{R}^p$ 为有界闭区域，$\Omega_0=\{\boldsymbol{x}\in\Omega;E_x(\boldsymbol{x})=0\}$ 为误差函数 $E(\boldsymbol{x})$ 的稳定点集合。令 $\Omega_{0,s}\subset\mathbf{R}$ 是 Ω_0 在第 s 坐标轴上的投影 $(1\leqslant s\leqslant p)$，即

$$\Omega_{0,s}=\{x_s\in\mathbf{R};\boldsymbol{x}=(x_1,\cdots,x_s,\cdots,x_p)^{\mathrm{T}}\in\Omega_0\}. \tag{18}$$

这里 $s=1,2,\cdots,p$。为证明上述算法的确定型收敛性，需要如下假设：

(A1) 激活函数 g 连续可微，$g'(t)$ 一致有界且满足局部 Lipschitz 连续；

(A2) 网络权值 $\{\boldsymbol{w},\boldsymbol{V}\}$ 一致有界；

(A3) 基于光滑 $L_{1/2}$ 正则化逆向迭代算法产生的迭代序列 $\{\boldsymbol{x}^{k}\}_{k=0}^{\infty}$ 一致有界；

(A4) $\Omega_{0,s}$ 不包含任何内点 $s=1,2,\cdots,p$。

为证上述收敛性定理，首先给出两个引理。

引理 1 设函数 $q(x)\in C^1[a,b]$，$q'(x)$ Lipschitz 连续，K 为 Lipschitz 常数，则 $q'(x)$ 在 $[a,b]$ 上几乎处处可导，且

$$|q''(x)|\leqslant K,\ a.e.\ [a,b]. \tag{19}$$

进而，存在常数 $C>0$，使得

$$q(x)\leqslant q(x_0)+q'(x_0)(x-x_0)+C(x-x_0)^2,x_0,x\in[a,b]. \tag{20}$$

证明 因为 $q'(x)$ 在 $[a,b]$ 上 Lipschitz 连续，所以 $q'(x)$ 是绝对连续函数，进而 $q'(x)$ 在 $[a,b]$ 上几

乎处处存在导数 $q''(x)$。

设 x_1 是 $q'(x)$ 在 $[a,b]$ 上任一可导点,则由可导定义及 $q'(x)$ 满足 Lipschitz 条件, 可知

$$|q''(x_1)|=\left|\lim_{h\to 0}\frac{q'(x_1+h)-q'(x_1)}{h}\right|=\lim_{h\to 0}\left|\frac{q'(x_1+h)-q'(x_1)}{h}\right|\leqslant K. \tag{21}$$

由 x_1 的选择, 可得

$$|q''(x)|\leqslant K,\ a.e.\ [a,b]. \tag{22}$$

利用积分型 Taylor 展式,

$$q(x)=q(x_0)+q'(x_0)(x-x_0)+(x-x_0)^2\int_0^1(1-t)q''(x_0+t(x-x_0))\,\mathrm{d}t\leqslant q(x_0)+q'(x_0)(x-x_0)+(x-x_0)^2\int_0^1 K(1-t)\,\mathrm{d}t=q(x_0)+q'(x_0)(x-x_0)+C(x-x_0)^2,\ C=\frac{K}{2},\ x_0,x\in[a,b]. \tag{23}$$

结论得证。

引理 2　数列 $\{b_m\}$ 有界且 $\lim\limits_{m\to\infty}(b_{m+1}-b_m)=0$,令 $\gamma_1=\liminf\limits_{n\to\infty}\limits_{m>n}b_m$, $\gamma_2=\limsup\limits_{n\to\infty}\limits_{m>n}b_m$,若记 $S=\{a\in\mathbf{R}:b_{i_k}\to a\ (k\to\infty)\}$,这里 b_{i_k} 是 b_m 的一个子列,则有 $S=[\gamma_1,\gamma_2]$。

证明　显然 $\gamma_1\leqslant\gamma_2$,且有 $S\subseteq[\gamma_1,\gamma_2]$。如果 $\gamma_1=\gamma_2$,那么 $\lim\limits_{m\to\infty}b_m=\gamma_1=\gamma_2$,命题 $S=[\gamma_1,\gamma_2]$ 成立。现在考虑 $\gamma_1<\gamma_2$ 的情形,只需要再证明 $S\supseteq[\gamma_1,\gamma_2]$。任取 $a\in(\gamma_1,\gamma_2)$,总可取 $\varepsilon>0$ 充分小,使得 $(a-\varepsilon,a+\varepsilon)\subset(\gamma_1,\gamma_2)$。由 $\lim\limits_{m\to\infty}(b_{m+1}-b_m)=0$,注意到对于所有充分大的 m,b_m 以越来越小的步伐无穷次往返于 γ_1 和 γ_2 之间,于是 $\{b_m\}$ 存在子列 b_{i_k},使得 $b_{i_k}\to a,(k\to\infty)$,从而 $a\in S$,因此 $(\gamma_1,\gamma_2)\subseteq S$。所以,$[\gamma_1,\gamma_2]\subseteq S$。结论得证。

根据上述两个引理,可以得出下述定理。

定理 1　(单调性) 若假设(A1)、(A2)、A(3)均成立,且学习率 η 满足式(36),对于任意给定的初始输入向量 $\boldsymbol{x}^0\in\mathbf{R}^p$,误差函数序列单调递减,即

$$E(\boldsymbol{x}^{k+1})\leqslant E(\boldsymbol{x}^k),\ k=0,1,2,\cdots, \tag{24}$$

并且存在 $E^*\geqslant 0$ 使得

$$\lim_{k\to\infty}E(\boldsymbol{x}^k)=E^*. \tag{25}$$

证明　由条件(A1),设

$$|g(t)|,|g'(t)|,|g''(t)|<\tilde{C}. \tag{26}$$

任意的 $t\in\mathbf{R}$,$\tilde{C}$ 为常数。由条件(A2),设

$$\|\boldsymbol{w}\|\leqslant C_1,\ \|\boldsymbol{v}_i\|\leqslant C_2. \tag{27}$$

$i=1,2,\cdots,n$,C_1,C_2 为常数,可以得到

$$\|\boldsymbol{\psi}^k\|=\|\boldsymbol{G}^{k+1}-\boldsymbol{G}^k\|=\sqrt{\sum_{i=1}^n(g(\boldsymbol{v}_i\cdot\boldsymbol{x}^{k+1})-g(\boldsymbol{v}_i\cdot\boldsymbol{x}^k))^2}=\sqrt{\sum_{i=1}^n(g'(t_i))^2\|\boldsymbol{v}_i\|^2\|\Delta\boldsymbol{x}^k\|^2}\leqslant\sqrt{n}\max_{1\leqslant i\leqslant n}g'(t_i)\|\boldsymbol{v}_i\|\|\Delta\boldsymbol{x}^k\|\leqslant\sqrt{n}\tilde{C}C_2\|\Delta\boldsymbol{x}^k\|. \tag{28}$$

其中

$$t_i=\boldsymbol{v}_i\cdot\Delta\boldsymbol{x}^{k+1}+\theta_i\boldsymbol{v}_i\cdot\Delta\boldsymbol{x}^k,\theta_i\in(0,1).$$

由积分型 Taylor 展式及误差函数定义,有

$$E(\boldsymbol{x}^{k+1})-E(\boldsymbol{x}^k)=[\tilde{g}(\boldsymbol{w}\cdot G(\boldsymbol{V}\boldsymbol{x}^{k+1}))-\tilde{g}(\boldsymbol{w}\cdot G(\boldsymbol{V}\boldsymbol{x}^k))]+\lambda[\|f(\boldsymbol{x}^{k+1})\|_{1/2}^{1/2}-\|f(\boldsymbol{x}^k)\|_{1/2}^{1/2}]=\tilde{g}'(\boldsymbol{w}\cdot G(\boldsymbol{V}\boldsymbol{x}^k))\boldsymbol{w}\cdot\boldsymbol{\psi}^k+(\boldsymbol{w}\cdot\boldsymbol{\psi}^k)^2\int_0^1(1-t)\tilde{g}''(\boldsymbol{w}\cdot G(\boldsymbol{V}\boldsymbol{x}^k)+t\boldsymbol{\psi}^k)\,\mathrm{d}t+\lambda\sum_{j=1}^p\sqrt{f(x_j^{k+1})}-\sqrt{f(x_j^k)}=\delta_1+\delta_2+\delta_3. \tag{29}$$

其中

$$\delta_1=\tilde{g}'(\boldsymbol{w}\cdot G(\boldsymbol{V}\boldsymbol{x}^k))\boldsymbol{w}\cdot\boldsymbol{\psi}^k=\sum_{i=1}^n\tilde{g}'(\boldsymbol{w}\cdot G(\boldsymbol{V}\boldsymbol{x}^k))w_ig'(\boldsymbol{v}_i\cdot\boldsymbol{x}^k)\boldsymbol{v}_i\cdot\Delta\boldsymbol{x}^k+\sum_{i=1}^n\tilde{g}'(\boldsymbol{w}\cdot G(\boldsymbol{V}\boldsymbol{x}^k))w_i\left(\int_0^1(1-t)g''(\boldsymbol{v}_i\cdot\boldsymbol{x}^k+t\boldsymbol{v}_i\cdot\Delta\boldsymbol{x}^k)\,\mathrm{d}t\right)(\boldsymbol{v}_i\cdot\Delta\boldsymbol{x}^k)^2. \tag{30}$$

由式(14) 及式(30) 得

$$\delta_1=-\frac{1}{\eta}\Delta\boldsymbol{x}^k\cdot\Delta\boldsymbol{x}^k-\lambda\nabla(\|f(\boldsymbol{x})\|_{1/2}^{1/2})\cdot\Delta\boldsymbol{x}^k+\sum_{i=1}^n\tilde{g}'(\boldsymbol{w}\cdot G(\boldsymbol{V}\boldsymbol{x}^k))w_i\left(\int_0^1(1-t)g''(\boldsymbol{v}_i\cdot\boldsymbol{x}^k+t\boldsymbol{v}_i\cdot\Delta\boldsymbol{x}^k)\,\mathrm{d}t\right)(\boldsymbol{v}_i\cdot\Delta\boldsymbol{x}^k)^2=-\frac{1}{\eta}\|\Delta\boldsymbol{x}^k\|^2-\lambda\sum_{j=1}^p\frac{f'(x_j^k)\Delta x_j^k}{2f^{1/2}(x_j^k)}+\sum_{i=1}^n\tilde{g}'(\boldsymbol{w}\cdot G(\boldsymbol{V}\boldsymbol{x}^k))\times w_i\left(\int_0^1(1-t)g''(\boldsymbol{v}_i\cdot\boldsymbol{x}^k+t\boldsymbol{v}_i\cdot\Delta\boldsymbol{x}^k)\,\mathrm{d}t\right)(\boldsymbol{v}_i\cdot\Delta\boldsymbol{x}^k)^2. \tag{31}$$

由式 (26) 和(27),可以得到

$$\delta_1\leqslant-\frac{1}{\eta}\|\Delta\boldsymbol{x}^k\|^2-\lambda\sum_{j=1}^p\frac{f'(x_j^k)\Delta x_j^k}{2f^{1/2}(x_j^k)}+A_1\|\Delta\boldsymbol{x}^k\|^2. \tag{32}$$

其中

$$A_1=\frac{1}{2}\tilde{C}^2C_1C_2^2.$$

$$\delta_2=(\boldsymbol{w}\cdot\boldsymbol{\psi}^k)^2\int_0^1(1-t)\tilde{g}''(\boldsymbol{w}\cdot G(\boldsymbol{V}\boldsymbol{x}^k)+t\boldsymbol{\psi}^k)\,\mathrm{d}t\leqslant\|\boldsymbol{w}\|^2\|\boldsymbol{\psi}^k\|^2A'_2\leqslant A''_2\|\boldsymbol{\psi}^k\|^2. \tag{33}$$

其中

$$A'_2=\frac{1}{2}\tilde{C},A''_2=A'_2C_1^2.$$

由式(28)可得

$$\delta_2\leqslant A_2\|\Delta\boldsymbol{x}^k\|^2. \tag{34}$$

其中 $A_2=n\tilde{C}^2C_2^2A''_2$。进一步

$$\delta_3=\lambda\sum_{j=1}^{p}(f^{1/2}(x_j^{k+1})-f^{1/2}(x_j^k))\leqslant \lambda\sum_{j=1}^{p}\frac{f'(x_j^k)\Delta x_j^k}{2f^{1/2}(x_j^k)}+\lambda A_3\|\Delta x^k\|^2. \tag{35}$$

其中 $t=x^{k+1}+\theta x_j^k,\theta\in(0,1)$。

综上,令

$$\eta<\frac{1}{A_4+\lambda A_3}, \tag{36}$$

可得

$$E(\boldsymbol{x}^{k+1})-E(\boldsymbol{x}^k)=\delta_1+\delta_2+\delta_3\leqslant-\left(\frac{1}{\eta}-A_4-\lambda A_3\right)\|\Delta\boldsymbol{x}^k\|^2\leqslant 0. \tag{37}$$

其中 $A_4=A_1+A_2$。误差函数单调性结论得证。

由 $E(\boldsymbol{x}^k)\geqslant 0,k\in\mathbf{N}$ 且 $E(\boldsymbol{x}^0)$ 有界,故存在 $E^*\geqslant 0$,满足

$$\lim_{k\to\infty}E(\boldsymbol{x}^k)=E^*. \tag{38}$$

定理 2 (弱收敛) 假设条件(A1)、(A2)、(A3)均成立,误差函数见式(8),并且学习率满足式(36),对任意给定初始输入向量 $\boldsymbol{x}^0\in\mathbf{R}^p$,$\boldsymbol{x}^k$ 由式(14)确定,那么

$$\lim_{k\to\infty}\|E_x(\boldsymbol{x}^k)\|=0. \tag{39}$$

证明 由定理 1 的结论,令 $\alpha=\frac{1}{\eta}-A_4-\lambda A_3$,有

$$E(\boldsymbol{x}^{k+1})\leqslant E(\boldsymbol{x}^k)-\alpha\|\Delta\boldsymbol{x}^k\|^2\leqslant E(\boldsymbol{x}^{k-1})-\alpha(\|\Delta\boldsymbol{x}^k\|^2+\|\Delta\boldsymbol{x}^{k-1}\|^2)\leqslant\cdots\leqslant E(\boldsymbol{x}^0)-\alpha\sum_{l=0}^{k}\|\Delta\boldsymbol{x}^k\|^2. \tag{40}$$

因为 $E(\boldsymbol{x}^k)\geqslant 0,k\in\mathbf{N}$,给定整数 $K\in\mathbf{N}^+$,有

$$\alpha\sum_{k=0}^{K}\|\Delta\boldsymbol{x}^k\|^2\leqslant E(\boldsymbol{x}^0). \tag{41}$$

令 $K\to\infty$,可以推出

$$\sum_{k=0}^{\infty}\|\Delta\boldsymbol{x}^k\|^2\leqslant\frac{1}{\alpha}E(\boldsymbol{x}^0)<\infty. \tag{42}$$

由级数收敛的必要性条件,下式成立:

$$\lim_{k\to\infty}\|\Delta\boldsymbol{x}^k\|=0. \tag{43}$$

再利用式(14)、(15),可以得到

$$\lim_{k\to\infty}\|E_x(\boldsymbol{x}^k)\|=0. \tag{44}$$

算法弱收敛性结论得证。

定理 3 (强收敛) 如果除了定理 1 的条件之外,条件(A4)也成立,可得到强收敛性结果,即存在 $\boldsymbol{x}^*\in\Omega_0$,使得

$$\lim_{k\to\infty}\boldsymbol{x}^k=\boldsymbol{x}^*. \tag{45}$$

证明 由条件(A3)可知,序列 $\{\boldsymbol{x}^k\}(k\in\mathbf{N})$ 必有收敛子列,其收敛点 $\boldsymbol{x}^*$ 必属于 Ω_0。不妨设 $\boldsymbol{x}^{k_i}\to\boldsymbol{x}^*(k_i\to\infty)$。由 $E_x(\boldsymbol{x})$ 的连续性

$$\|E_x(\boldsymbol{x}^*)\|=\lim_{i\to\infty}\|E_x(\boldsymbol{x}^{k_i})\|=\lim_{m\to\infty}\|E_x(\boldsymbol{x}^k)\|=0. \tag{46}$$

则 $\boldsymbol{x}^*$ 是 $E(\boldsymbol{x})$ 的一个稳定点,因此 $\{\boldsymbol{x}^k\}$ 至少有一个聚点且任一聚点都是 $E(\boldsymbol{x})$ 的稳定点。

用反证法证明 $\{\boldsymbol{x}^k\}$ 只有一个聚点。若 $\{\boldsymbol{x}^k\}$ 有两个不同的聚点:$\bar{\boldsymbol{x}}$,$\tilde{\boldsymbol{x}}$。由式(15)可知 $\lim_{k\to\infty}|x_j^{k+1}-x_j^k|=0,j=1,2,\cdots,p$ 成立。不失一般性,假设对应的第一个分量不相等,即 $\bar{x}_1\neq\tilde{x}_1$,则对 $\forall\lambda\in(0,1)$,令 $x'_1=\lambda\bar{x}_1+(1-\lambda)\tilde{x}_1$,由引理 2,存在序列 $\{x_1^{k_{i_1}}\}\subset\{x_1^k\}$ 使得 $x_1^{k_{i_1}}\to x'_1(i_1\to\infty)$ 由序列 $\{x^{k_{i_1}}\}$ 的有界性,可知存在子序列 $\{k_{i_2}\}\subset\{k_{i_1}\}$,满足 $x^{k_{i_2}}\to x'_2(i_2\to\infty)$,以此进行下去,可以得到 $\boldsymbol{x}^{k_{i_t}}\to x'_t(i_t\to\infty)$,$t=1,2,\cdots,p$,记 $x'=\{x'_1,x'_2,\cdots,x'_p\}$,所以 $\boldsymbol{x}'\in\Omega_0$,即对于任意的 $\lambda\in(0,1)$,x' 是 $\{\boldsymbol{x}_k\}$ 的一个聚点。这与假设(A4)矛盾,所以 $\boldsymbol{x}^*$ 必是 $\boldsymbol{x}^k$ 的唯一聚点,强收敛性得证。

3 数值试验

文献[17]给出了黄河下游灌区 1983—1995 年灌溉引水量与降雨量、灌溉面积以及河南、山东两地灌溉定额数据(表 1)。本实验分为两个步骤:①分别利用带 $L_{1/2}$ 正则化惩罚项前馈神经网络和普通前馈神经网络(BPNN)构建引水量预测模型;②利用训练完成的神经网络权值,建立本文中提出的逆向迭代神经网络模型,给出误差函数变化曲线图以及误差函数关于输入向量的梯度范数变化趋势图,最后得到各因素对正常灌溉引水量的影响程度。

为验证和对比带 $L_{1/2}$ 正则项前馈神经网络和普通前馈神经网络的精度,选取前 10 年数据作为训练样本,后 3 年数据作为测试样本。为保持量纲一致性,对所有数据进行归一化处理。根据研究问题,选用 3 层神经网络,输入层 5 个输入节点(4 个代表影响因子,1 个为阈值);隐层激活函数选用 Sigmoid 函数,节点数为 5 个;输出层 1 个节点,为灌溉引水量。取学习率为 0.3,初始权值和阈值在区间(-0.5,0.5)随机选取,误差限设为 0.05,惩罚项系数取 0.16。经过网络训练对比,得到结果见表 2。可以看到文献

[17]采用的是 4-6-1 的 BP 神经网络结构,对学习样本达到很高的计算精度,但是预测误差明显偏大,其主要原因是训练过度造成的过拟合现象。通过选取合适的网络结构和参数,带 $L_{1/2}$ 正则项前馈神经网络可以很好地模拟学习样本,同时也能正确地反映预测样本的变化趋势,具有较强的泛化能力。

表 1　黄河下游正常灌区灌溉饮水量与影响因素

Table 1　Normal irrigation water quantity of lower Yellow River and influence factors

年份	平均降水量/mm	正常灌溉面积/10^4 hm^2	河南灌区灌溉定额/(m^3 · hm^{-2})	山东灌区灌溉定额/(m^3 · hm^{-2})	正常灌溉引水量/10^8 m^3
1983	596	101.3	8 355	6 150	67.7
1984	704	138.1	7 785	418	66.8
1985	630	134.1	7 410	4 065	58.8
1986	381	145.4	8 490	5 670	89.1
1987	544	150.4	9 780	5 055	81.6
1988	432	156.4	9 165	5 280	89.8
1989	460	174.2	7 050	6 900	120.7
1990	850	166.2	8 355	4 845	85.2
1991	569	195.3	7 830	4 405	85.6
1992	514	202.7	7 080	5 100	100.6
1993	632	221.5	7 185	4 155	93.2
1994	694	199.6	6 045	4 380	79.3
1995	615	192.3	6 150	4 455	79.9

表 2　带正则项前馈神经网络与普通前馈神经网络计算结果对比

Table 2　Comparison of calculated results with regularization of feed-forward neural network and ordinary feedforward neural network

样本类型	年份	实际引水量/10^8 m^3	$L_{1/2}$ BPNN 计算引水量/10^8 m^3	$L_{1/2}$ BPNN 法相对误差/%	BPNN 法计算引水量/10^8 m^3	BPNN 法相对误差/%
训练集	1983	67.7	68.25	0.812	67.7	0.00
	1984	66.8	65.77	-1.542	66.8	0.00
	1985	58.8	59.89	1.854	58.8	0.00
	1986	89.1	88.57	-0.595	89.1	0.00
	1987	81.6	81.63	0.037	81.6	0.00
	1988	89.8	89.83	0.033	89.8	0.00
	1989	120.7	119.94	-0.630	120.7	0.00
	1990	85.2	85.37	0.199	85.2	0.00
	1991	85.6	85.60	0.000	85.6	0.00
	1992	100.6	101.14	0.537	100.6	0.00
测试集	1993	93.2	91.5	-1.824	86.2	-7.51
	1994	79.3	80.3	1.261	81.3	2.52
	1995	79.9	79.5	-0.501	79.4	-0.63

在完成上述网络学习的基础上,固定得到的训练权值和阈值,采用逆向迭代算法训练神经网络,学习率取为 0.4,惩罚项系数为 0.15,误差限设为 0.001,最大迭代次数 7 000。图 2 显示误差函数曲线随迭代次数的增加单调递减,验证了本文定理 1 的结论;图 3 误差函数关于输入向量的梯度范数变化曲线随着迭代次数的增加,梯度范数趋于零,也很好地验证了带 $L_{1/2}$ 正则项逆向神经网络迭代算法的弱收敛性结论(定理 2)。最后,从表 1 选取 1994 年的数据验证带 $L_{1/2}$ 正则项神经网络网络逆向迭代算法的有效性,计算结果为:615.45,212.7,5 989.2,4 310.5。结果表明,除平均降水量偏差较大之外,其余均能正常反应当年的实际数据,这同网络参数的选择以及学习率、惩罚项系数都有较大关系。

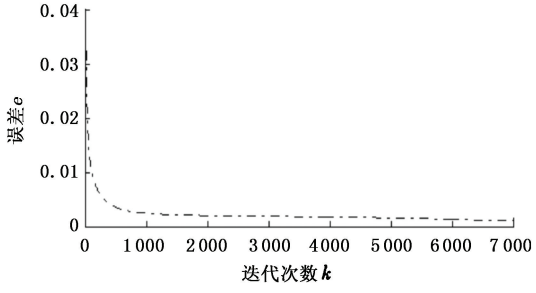

图 2　带 $L_{1/2}$ 正则项逆向神经网络迭代算法误差函数曲线

Fig. 2　Curve of error function for inverse iterative neural networks with $L_{1/2}$ penalty

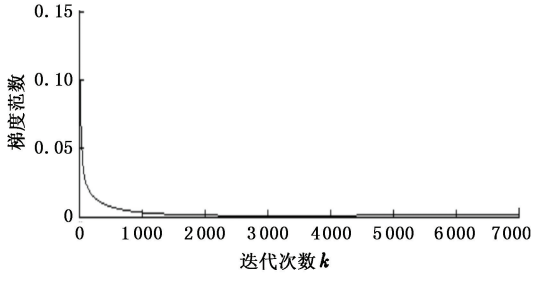

图 3　带 $L_{1/2}$ 正则项逆向神经网络迭代算法弱收敛示意图

Fig. 3　Performance of weak convergence for inverse iterative neural networks with $L_{1/2}$ penalty

4　结束语

从改善网络稀疏性角度出发,首先给出带 $L_{1/2}$ 正则项的神经网络逆向迭代算法的迭代公式。由于 $L_{1/2}$ 正则子非凸、非光滑,并且不满足非扩张性质,容易导致网络训练过程中出现数值振荡与不稳定,并且难于进行理论分析。为了克服这一缺点,新算法用特定的光滑函数逼近 $L_{1/2}$ 正则子,引入了带光滑 $L_{1/2}$ 正则项的神经网络逆向迭代算法,证明了学习率在满足特定条件下,误差函数是单调递减的,并严格证明了该算法的弱(强)收敛性。最后通过黄河灌溉引水量预测算例进一步验证了本文算法的有效性和理论结果。

参考文献:

[1] ZURADA J M. Introduction to artificial neural systems [M]. Minnesota: West Publishing Company St Paul, 1992.

[2] HAYKIN S S. Neural networks and learning machines [M]. New Jersey: Pearson Education, Inc. Upper Saddle River, 2009.

[3] REED R. Pruning algorithms-a survey[J]. Neural Networks, IEEE Transactions on, 1993,4(5):740-747.

[4] ISHIKAWA M. Structural learning with forgetting[J]. Neural Networks, 1996,9(3):509-521.

[5] SETIONO R. A penalty function approach for pruning feedforward neural networks [J]. Neural Computation, 1997,9(1):185-204.

[6] SHAO H M, WU W, LIU L J. Convergence of online gradient method with penalty for BP neural networks J]. Communications in Mathematical Research, 2010, 26(1):67-75.

[7] WANG J, YANG J, WU W. Convergence of cyclic and almost-cyclic learning with momentum for feedforward neural networks[J]. Neural Networks, IEEE Transactions on, 2011,22(8):1297-1306.

[8] UHLMANN G. Inverse problems and applications: Inside out II[M]. Cambridge: Cambridge University Press, 2003.

[9] ZAMPARO M, STRAMAGLIA S, BANAVAR J, et al. Inverse problem for multivariate time series using dynamical latent variables[J]. Physica A: Statistical Mechanics and its Applications, 2012,391(11):3159-3169.

[10] KINDERMANN J, LINDEN A. Inversion of neural networks by gradient descent [J]. Parallel Computing, 1990,14(3):277-286.

[11] FANNI A, MONTISCI A. A neural inverse problem approach for optimal design[J]. Magnetics, IEEE Transactions on, 2003,39(3):1305-1308.

[12] CHERUBINI D, FANNI A, MONTISCI A, et al. Inversion of MLP neural networks for direct solution of inverse problems[J]. Magnetics, IEEE Transactions on, 2005, 41(5):1784-1787.

[13] HAYAKAWA Y, NAKAJIMA K. Design of the inverse function delayed neural network for solving combinatorial optimization problems [J]. Neural Networks, IEEE Transactions on, 2010,21(2):224-237.

[14] DUREN R W, MARKS R J, REYNOLDS P D, et al. Real-time neural network inversion on the SRC-6e reconfigurable computer [J]. Neural Networks, IEEE Transactions on, 2007,18(3):889-901.

[15] 孟少奇. 神经网络逆向迭代算法的收敛性[D]. 大连:大连理工大学数学科学学院, 2007.
MENG Shaoqi. Convergence of an inverse iteration algorithm for neural networks[D]. Dalian: School of Mathematical Sciences, Dalian University of Technology, 2007.

[16] XU Z B, ZHANG H, WANG Y, CHANG X Y, et al. $L_{1/2}$ regularization[J]. Science China-Information Sciences, 2010,53(6):1159-1169.

[17] 黄国如, 胡和平. 基于 BP 神经网络的黄河下游引黄灌区引水量分析[J]. 灌溉排水, 2000,19(3):20-23.
HUANG Guoru, HU Heping. Analyzing water diversion demand for irrigation areas at lower reach of Yellow River with BP neural network techniques [J]. Journal of Irrigation and Drainage, 2000,19(3):20-23.

(编辑 修荣荣)