综合反向传播算法

王科俊 金鸿章 李国斌
(哈尔滨工程大学自动化学院·哈尔滨,150001)

摘要:提出一种用于多层前向神经网络的综合反向传播算法.该算法使用了综合考虑绝对误差和相对误差的广义指标函数,采用了在网络输出空间搜索的反传技术,具有动态自调整学习率和动量因子,有神经元激活特性自调整、减少平台现象和消除学习过程中不平衡现象的能力.对比实验表明该算法有比基本 BP 算法快得多的收敛速度,并能取得全局最优解.

关键词:神经网络;学习算法;广义指标函数

A Synthetically Backpropagation Algorithm

Wang Kejun, Jin Hongzhang and Li Guobin
(Automation College, Harbin Engineering University · Harbin, 150001, P.R.China)

Abstract: This paper presents a synthetically backpropagation algorithm for multilayered forward neural networks. A new general index function that consider the effect of absolute error and relative error on NN learing and performance and the backpropagation technique based on searching output space are proposed and used in the algorithm. The algorithm has both a dynamical adaptive regulation learning rate and a variable momentum coefficient, and has ability of self regulation active characteristic, eliminating flat phenomenon and convergence no equilibrium phenomenon during training. The contrast experiments indicate that the algorithm has more fast convergence speed than BP algorithm and can achieve a global optimal solution.

Key words: neural network; learning algorithm; general index function

1 引言(Introduction)

误差反向传播算法(BP)[1]是用于多层神经网络训练的著名算法,有理论依据坚实、推导过程严谨、物理概念清楚、通用性强等优点.但是,人们在使用中发现 BP 算法存在收敛速度缓慢、易陷入局部极小等缺点.

本文综合考虑网络的泛化能力、训练的快速性和全局最优性,提出一种多层前向神经网络的改进训练算法——综合反向传播算法.

2 广义误差指标函数(General error index function)

算法中采用了如下作者提出的广义误差函数:

$$E(\lambda)=\sum_p E_p(\lambda)=\sum_{p=1}^{P}\sum_{i=1}^{N_L}\{d_i^{(p)}(d_i^{(p)}-y_i^{(p)(L)}(t))+\frac{\lambda}{2}[(y_i^{(p)(L)}(t))^2-(d_i^{(p)})^2]+\frac{1}{2}\left(1-\frac{y_i^{(p)(L)}(t)}{d_i^{(p)}+\varepsilon}\right)^2\}. \quad (1)$$

其中 $d^{(p)}i, y_i^{(p)(L)}(t)$ 分别指在第 p 个训练模式下网络输出层第 i 个输出神经元的期望输出和实际输出;P 指训练模式总数;N_L 指网络输出层神经元数;λ 为可变因子,在训练期间从 1 递减到 0;ε 为很小的正数,用于防止期望响应 $d_i=0$,导致被零除的现象出现.

(1)式由 Karayiannis 的广义指标函数[2]和相对误差项构成,它综合考虑了反传算法的收敛速度和学习精度,是绝对误差和相对误差相结合的一种指标函数.提出并采用这种指标函数的原因是广义指标函数((1)式中的前二项)能提高算法的收敛速度[2],相对误差能提高算法的精度[3].

使用中可变因子 λ 可由训练过程的总误差调整,如取 $\lambda=\exp(-r/E^2)$,r 为正实数.显然,在学习开始阶段 $E\gg 1,\lambda\approx 1$,在训练过程中随着 E 的减小,λ 逐渐减小,当 $E\ll 1$ 时,$\lambda=0$.

3 基于输出空间搜索的反传技术(Backpropagation technique based on searching output space)

本文于 1997 年 4 月 23 日收到.1998 年 9 月 18 月收到修改稿.

根据最优化理论的结论:如果某一函数在某一空间中是凸函数,那么函数的局部极小点也是全局最小点.因此如果BP算法的指标函数在某一空间是凸函数,那么在此空间的极值即是全局最小.观察基本BP算法的指标函数 $E = \sum_{p=1}^{P}\sum_{i=1}^{N_L}(d_i^{(p)} - y_i^{(p)(L)}(t))^2/2$,显然它在权空间不是凸函数,而在输出空间是凸函数.因此如将权更新过程改在输出空间上进行,必须得到全局最优解.

为说明本文算法具有全局最优性,必须证明(1)式的广义误差指标函数是输出空间的凸函数.

3.1 广义误差指标函数的凸性(Convexity of general error index function)

定义 3.1[4] 设凸集 $D \subseteq \mathbb{R}^n$,若 $\forall x_1, x_2 \in D$, $\forall \alpha \in (0,1)$ 恒有

$$f(\alpha x_1 + (1-\alpha)x_2) \leqslant \alpha f(x_1) + (1-\alpha)f(x_2), \tag{2}$$

则称 $f(x)$ 为在 D 上的凸函数.

如恒有

$$f(\alpha x_1 + (1-\alpha)x_2) < \alpha f(x_1) + (1-\alpha)f(x_2), \tag{3}$$

则称 $f(x)$ 为在 D 上的严格凸函数.

定义 3.2[4] 设有规则

$$(\mathrm{P}) \qquad \begin{cases} \min f(x), \\ \text{s.t.}\, g_i(x) \geqslant 0,\ i = 1,2,\cdots,m. \end{cases} \tag{4}$$

当 $f(x)$ 及 $-g_i(x)(i = 1,2,\cdots,m)$ 都是严格凸函数,称规则(P)为凸规则.

定理 3.1[4] 设 $f(x)$ 为严格凸函数,$-g_i(x)(i = 1,2,\cdots,m)$ 都为凸函数,(P)的最优解集合 $R^* \neq \Phi$,则(P)的最优解必唯一.

定理 3.2 最小二乘指标函数 $E = \sum_{p=1}^{P}\sum_{i=1}^{N_L}(d_i^{(p)} - y_i^{(p)(L)}(t))/2$ 是输出空间上的严格凸函数.

定理3.2是显然成立的.由于在输出空间上进行优化,神经网络的输出即为被优化的对象,因此是无约束优化.根据定理3.1,此时得到的解即为唯一的最优解.对(1)式的广义误差指标函数 $E(\lambda)$ 有定理3.3.

定理 3.3 广义误差指标函数 $E(\lambda)$ 是输出空间上的严格凸函数.

证 为简单,仅考虑在一个模式下的误差,将 $d_i^{(p)}$ 简记为 d_i,$y_i^{(p)(L)}(t)$ 记为 y_i;ε 很小忽略之,则

$$E_p(\lambda) = \sum_{i=1}^{N_L}\left[\left(\frac{\lambda}{2} + \frac{1}{2d_i^2}\right)(d_i - y_i)^2 + (1-\lambda)d_i(d_i - y_i)\right].$$

令

$$X^{\mathrm{T}} = [d_1 - y_1, d_2 - y_2, \cdots, d_{N_L} - y_{N_L}],$$

$$D^{\mathrm{T}} = [d_1, d_2, \cdots, d_{N_L}],$$

$$Y^{\mathrm{T}} = [y_1, y_2, \cdots, y_{N_L}],$$

$$B = \begin{bmatrix} \lambda + \dfrac{1}{d_1^2} & & 0 \\ & \ddots & \\ 0 & & \lambda + \dfrac{1}{d_{N_L}^2} \end{bmatrix},$$

则 $X^{\mathrm{T}} = (D - Y)^{\mathrm{T}}$,故

$$E_p(X) = \frac{1}{2}X^{\mathrm{T}}BX + (1-\lambda)D^{\mathrm{T}}X.$$

若对输出空间 $D \subseteq \mathbb{R}^{N_L}$,有 $\forall X_1, X_2 \in D$, $\forall \alpha \in (0,1)$,则经推导有

$$E_p[\alpha X_1 - (1-\alpha)X_2] - \alpha E_p(X_1) - (1-\alpha)E_p(X_2) = -\frac{1}{2}\alpha(1-\alpha)(X_1 - X_2)^{\mathrm{T}}B(X_1 - X_2),$$

由 B 阵的定义知它为正定阵,故上式 <0,则

$$E_p[\alpha X_1 + (1-\alpha)X_2] < \alpha E_p(X_1) + (1-\alpha)E_p(X_2).$$

根据定义3.1,$E_p(X)$ 为输出空间 D 上的严格凸函数,进而可知 $E(\lambda)$ 亦为 D 上的严格凸函数.

证毕.

3.2 基于输出空间搜索的全局最优反传算法(A global optimal backpropagation algorithm based on searching output space)

参考定理3.3的证明过程,把(1)式的广义误差函数重写如下:

$$E(\lambda) = \sum_{p=1}^{P}E_p(\lambda) = \sum_{p=1}^{P}\left[\frac{1}{2}(D - Y)^{\mathrm{T}}B(D - Y) + (1-\lambda)D^{\mathrm{T}}(D - Y)\right]. \tag{5}$$

基于输出空间中的梯度下降方向改变输出 $Y^{\mathrm{T}} = [y_1, y_2, \cdots, y_{N_L}]$,即

$$\Delta Y(t) = Y(t+1) - Y(t) \approx -\eta \nabla_Y E_p(\lambda). \tag{6}$$

在神经网络权 A 的微小变动下,利用Taylor级数,将网络输出针对权的微小变化展开:

$$\Delta Y = Y(A+\Delta A, X) - Y(A, X) \approx$$

$$\nabla_A Y \cdot \Delta A + \frac{1}{2}\Delta A^{\mathrm{T}} \nabla_A^2 Y(A+\zeta\Delta A, X)\Delta A. \quad (7)$$

其中 $\zeta \in (0,1)$, X 为网络输入.

采用一阶逼近,由(6)式和(7)式得:

$$\Delta A = -(\nabla_A Y(t))^{-1} \cdot \eta \cdot \nabla_Y E_p(\lambda). \quad (8)$$

这里需求 $\nabla_A Y(t)$ 的伪逆,这在计算上是极其困难的.但是,注意到多层前向网络的特殊结构,某一输出神经元 i 的权与其它输出神经元的权无关.那么 ΔY 可写为

$$\Delta Y = [\nabla_{A_H} Y, \nabla_{A_{Y_1}} Y, \cdots, \nabla_{A_{Y_{N_L}}} Y] \begin{bmatrix} \Delta A_H \\ \Delta A_{Y_1} \\ \vdots \\ \Delta A_{Y_{N_L}} \end{bmatrix}. \quad (9)$$

其中 A_H 指隐层中的权, A_{Y_i} 为与输出神经元 i 相连的权. ΔY 的分量为:

$$\Delta y_i = \nabla_{A_H} y_i \cdot \Delta A_H + \nabla_{A_{y_i}} y_i \cdot \Delta A_{y_i} =$$

$$\nabla_{A^i} y_i \Delta A^i, \ i = 1,2,\cdots,N_L. \quad (10)$$

其中 A^i 指与输出 y_i 相关的所有权.

(10)式表明 Δy_i 是 $\nabla_{A^i} y_i$ 与 ΔA^i 的内积,如果我们在 $\nabla_{A^i} y_i$ 的方向上选取 ΔA^i,那么内积最大,即

$$\Delta y_i = \| \nabla_{A^i} y_i \| \cdot \| \Delta A^i \|, \ i = 1,2,\cdots,N_L. \quad (11)$$

设 ΔA^i 的正则化分量与 $\nabla_{A^i} y_i$ 的正则化分量相等,即

$$\Delta A_k = \| \Delta A^i \| \cdot \frac{\partial y_i / \partial A_k}{\| \nabla_{A^i} \cdot y_i \|}. \quad (12)$$

其中 ΔA_k 是网络中任意一个权的变化量.

考虑到(6)式和(7)式,对一个训练模式有

$$\Delta A_k = -\eta \nabla_{y_i} E_p(\lambda) \frac{\partial y_i}{\partial A_k} \Big/ \| \nabla_{A^i} y_i \|^2. \quad (13)$$

式中 $\| \nabla_{A^i} y_i \|^2$ 是针对与输出层神经元 i 相关的所有权的范数.

多层前向网络可用(14)式描述:

$$\begin{cases} x_i^{(l+1)}(t) = \sum_{j=0}^{N_l} a_{ij}^{(l+1)} y_j^l(t), \\ y_i^{(l+1)}(t) = g_i^{(l+1)}(x_i^{(l+1)}(t)). \end{cases} \quad (14)$$

其中 $x_i^{(l+1)}, y_i^{(l+1)}$ 分别表示第 $l+1$ 层第 i 个神经元的状态和输出, $a_{ij}^{(l+1)}$ 表示第 $l+1$ 层第 i 个神经元与第 l 层第 j 个神经元间的连接权, $a_{io}^{(l+1)}$ 表示第 $l+1$ 层第 i 个神经元的阈值, $g_i^{(l+1)}(\cdot)$ 表示第 $l+1$ 层第 i 个神经元的激活函数. l 表示层数,取 $0,1,\cdots,L-1$.

考虑到(14)式(且 $y_0^{(l)} = 1$), $\| \nabla_{A^i} y_i \|^2$ 可写为(欧氏空间):

$$\| \nabla_{A^i} y_i^{(L)} \|^2 =$$

$$\sum \left(\frac{\partial y_i^{(L)}}{\partial A_j} \right)^2 =$$

$$(g_i^{(L)'}(x_i^{(L)}))^2 [(y_0^{(L-1)})^2 + \sum_{j=1}^{N_{L-1}} [(y_j^{(L-1)})^2 +$$

$$(a_{ij}^{(L)} \cdot g_j^{(L-1)'}(x_j^{(L-1)}))^2 \cdot [(y_0^{(L-2)})^2 +$$

$$\sum_{k=1}^{N_{L-2}} [(y_k^{(L-2)})^2 + (a_{jk}^{(L-1)} \cdot g_k^{(L-2)'}(x_k^{(L-2)}))^2 \cdot$$

$$[(y_0^{(L-3)})^2 + \sum_{m=1}^{N_{L-3}} [(y_m^{(L-3)})^2 + \cdots]]]]]]. \quad (15)$$

尽管(15)式形式上非常复杂,但它们的计算可与网络的前向计算同步进行,计算量和存贮量的需求并不大.

定义 3.3 网络的广义误差信号为:

$$\delta_i^{(l)} = -\frac{\partial E_p(\lambda)}{\partial y_i^{(l)}} \Big/ \| \nabla_{A^k} y_k^{(L)} \|^2. \quad (16)$$

类似于 BP 算法的推导过程可以得到如下在输出空间搜索的反传算法计算公式:

$$a_{ij}^{(l)}(t+1) =$$

$$a_{ij}^{(l)}(t) + \eta \cdot \delta_i^{(l)} \cdot g_i^{(l)'}(x_i^{(l)}) \cdot y_j^{(l-1)}(t) +$$

$$\alpha \cdot (a_{ij}^{(l)}(t) - a_{ij}^{(l)}(t-1)),$$

$$\delta_i^{(l)}(t) =$$

$$\begin{cases} \left[\left(\lambda + \frac{1}{d_i^2} \right)(d_i - y_i^{(L)}) + (1-\lambda) d_i \right] \Big/ \| \nabla_{A^i} y_i^{(L)} \|^2, \\ \qquad \text{对输出层}, \\ \sum_{j=1}^{N_{l+1}} \delta_j^{(l+1)}(t) \cdot g_j^{(l+1)'}(x_j^{(l+1)}(t)) \cdot a_{ji}^{(l+1)}(t), \\ \qquad \text{对其它层}, \end{cases} \quad (17)$$

$$\| \nabla_{A^i} y_i^{(L)} \|^2 =$$

$$(g_i^{(L)'}(x_i^{(L)}))^2 [(y_0^{(L-1)})^2 + \sum_{j=1}^{N_{L-1}} [(y_j^{(L-1)})^2 +$$

$$(a_{ij}^{(L)} \cdot g_j^{(L-1)'}(x_j^{(L-1)}))^2 \cdot [(y_0^{(L-2)})^2 +$$

$$\sum_{k=1}^{N_{L-2}} [(y_k^{(L-2)})^2 + (a_{jk}^{(L-1)} \cdot g_k^{(L-2)'}(x_k^{(L-2)}))^2 \cdot$$

$$[(y_0^{(L-3)})^2 + \sum_{m=1}^{N_{L-3}} [(y_m^{(L-3)})^2 + \cdots]]]]]].$$

(17)式中的权更新公式与基本 BP 算法的权更新公式完全相同,但这里的广义误差的定义和计算却不相同,这一变化使原来在复杂的权空间的搜索转换为在简单的输出空间的搜索.广义误差指标函

数的凸性使这种搜索能保证得到全局最优解.

4 变学习率、动量因子的方法及在本文算法采用的其它措施(Methods of adaptive regulation learning rate, momentum coefficient, and other measures in the algorithm)

4.1 学习率 η、动量因子 α 的调整方法(The regulation methods of learning rate and momentum coefficient)

基于作者在文[5]中对学习率 η 和动量因子 α 方法的讨论,为加快收敛速度,在算法中采用了如下由作者提出的变学习率 η 方法:

$$\eta'(t) = e^{\beta}\cos\varphi \cdot \eta(t-1) + \gamma(E(t-1) - E(t)), \tag{18}$$

$$\eta(t) = \begin{cases} \eta_{\max}, & 若\ \eta'(t) > \eta_{\max}, \\ \eta_{\min}, & 若\ \eta'(t) < \eta_{\min}, \\ \eta'(t), & 其它. \end{cases} \tag{19}$$

其中 β,γ 为大于零的常数,可根据具体问题适当选取;$\cos\varphi$ 为超曲面的方向余弦[6];$\eta_{\max},\eta_{\min}$ 分别为最大学习率和最小学习率.

动量因子 α 的调整方法如下:

$$\alpha_j(t) = \delta_j^2(t) / \sum_{m=1}^{t-1} \delta_j^2(m). \tag{20}$$

其中 δ_j 为反传的广义误差信号.

4.2 减少平台现象的措施[5](The measures of eliminating flat phenomenon)

1) 网络输入自动正则化;

2) 网络中的神经元采用双曲正切函数为作用函数;

3) 网络中神经元的激活特性随训练自适应调整;

4) 网络中权、阈值的初值在[-0.5, +0.5]之间随机选取.

4.3 消除训练过程中收敛不平衡现象的方法[5](The methods of eliminating convergence on equilibium phenomenon during training)

4.3.1 跳跃学习(Jumping learning)

对模式不平衡,设定一模式收敛条件 $e^{p}_{\min}$,如某一输入样本模式下的输出误差 E_p 满足 $E_p < e^{p}_{\min}$,跳过对此模式的权、阈值修改过程.

对单元(神经元)收敛不平衡,设定权收敛条件 $e^{\delta}_{\min}$,如某一神经元在某一模式下的广义误差 δ 满足 $\delta < e^{\delta}_{\min}$,则跳过与此神经元相连的所有权、阈值的修正.

4.3.2 过滤误导修正(Filtering error correction)

比较连续两个模式下的模式输出误差,如果误差下降则修正权、阈值,否则跳过修正.

5 算法的实验研究(Experiment studies of the algorithm)

为了说明算法的性能,作者分别采用:1) 基本BP算法(输出层和隐层均采用 $\tanh(x)$ 为作用函数)(TT);2) 具有激活特性自调整的BP算法(BPT)[5]($g(x) = \lambda\tanh(\beta x) + \gamma$);3) 基于输出空间搜索的BP算法(OBP)(具有可调参数的作用函数 $g(x) = \lambda\tanh(\beta x) + \gamma$);4) 本文算法(SBP),进行了奇偶3问题的对比实验.

对比实验结果如表1所示,显然,本文算法在收敛所需的时间和循环次数上都优于其它方法,且受学习率和动量因子的初值影响很小.

表1 奇偶3问题对比实验结果 网络结构 3-3-1,$\alpha=0.01$,训练结束条件 MSE $< 10^{-6}$

Table 1 Contrast experiment results on parity-3 problem

η		0.05	0.075	0.10	0.25	0.50
TT	次数	10000	6689	5370	1400	350
	时间	4′45″07	3′11″34	2′33″59	局部极值	局部极值
	精度	1.9144×10^{-7}	9.998×10^{-8}	9.997×10^{-8}	8.688×10^{-2}	1.0033×10^{-1}
BPT	次数	1976	696	538	609	10000
	时间	1′53″15	40″10	31″68	35″17	9′30″84
	精度	9.958×10^{-8}	9.528×10^{-8}	9.447×10^{-8}	9.731×10^{-8}	2.473×10^{-7}
OBP	次数	1944	1343	815	563	824
	时间	1′51″28	1′07″00	46″91	32″52	48″27
	精度	9.8035×10^{-8}	9.97095×10^{-8}	9.82234×10^{-8}	9.8152×10^{-8}	9.9042×10^{-8}
SBP	次数	417	408	451	552	486
	时间	24″23	22″23	26″09	31″92	28″17
	精度	2.92428×10^{-8}	8.96773×10^{-8}	9.11332×10^{-8}	6.10537×10^{-8}	7.73872×10^{-8}

参考文献(References)

1 Rumelhart D E and McClelland D E. Parallel Distributed Processing. Cambridge, MA: MIT Press, 1986

2 Karayianais N B. Recent developments in supervised learing. IEEE Int. Conf. on Syst., Man, and Cybern., 1992, 1: 387 - 391

3 陆金桂等. 多层神经网络 BP 算法研究. 计算机工程, 1994, 20(1): 41 - 42

4 陈开周编著. 最优化计算方法. 西安: 西北电讯工程学院出版社, 1985

5 王科俊. 神经网络几个理论问题的研究及其在船舶横摇运动建模与预报中的应用: [博士学位论文]. 哈尔滨: 哈尔滨工程大学, 1995

6 Franzin M A. Speech recognition with backpropagation. Proc. of IEEE 9th Annual Conf. on Engineering in Medicine and Bielogy, 1987, 1702 - 1703

本文作者简介

王科俊 1962 年生. 1995 年在哈尔滨工程大学获博士学位. 现任哈尔滨工程大学教授. 主要研究方向为神经网络, 专家系统, 智能控制系统理论及应用等.

金鸿章 1946 年生. 1970 年毕业于哈尔滨军事工程学院. 哈尔滨工程大学教授, 博士生导师. 研究领域为智能控制, 船舶运动控制研究.

李国斌 1937 年生. 1962 年毕业于哈尔滨军事工程学院. 哈尔滨工程大学教授, 博士生导师. 主要从事船舶特辅装置与系统的设计、研究和控制工作.

通控博软荣获 1998 年最佳产品奖

被誉为工业控制和仪器仪表行业"奥斯卡"奖的《控制工程》(Control Engineering)杂志 1998 年最佳产品奖(Editor's Choice Award)最近揭晓. 美国通控集团博软分部(CyboSoft, General Cybernation Group Inc)的 CyboCon 先进控制软件荣获这一大奖. 在得奖的 6 种软件中, 还包括了微软公司(Microsoft)的 Windows CE 及霍尼韦尔(Honeywell)的 Porfit Suite 软件.

《控制工程》是世界工控和仪表行业的权威杂志. 每个月该杂志对数百种新产品进行评估. 选出其中最好的加以报道. 每年年底, 由该杂志的编辑们按照产品的技术先进性、对市场的影响力和对工业界的贡献三项标准选出该年度的最佳产品.

今年的发奖大会于 3 月 14 日在美国芝加哥举行. 无模型自适应控制技术的发明人, 通控集团总载程树行博士(Dr. George S. Cheng)等人代表该公司出席了颁奖仪式. 成立于 1994 年的美国通控集团公司近年来以其专有的无模型自适应控制(Model-Free Adaptive Control)技术和综合智能(Combined Intelligence)技术在国际工控舞台上迅速崛起. 其用户大都是美国五百强企业(Fortune 500)中的佼佼者.

CyboCon 是世界上首套"即插即用"式单变量和多变量控制软件, 可对简单或复杂的工业过程作自适应控制. 与传统的自适应控制技术相比, CyboCon 的无模型自适应(MFA)控制器的实用性非常强. 使用者无需进行控制器设计、过程辨识, 也不需知道过程的定量知识, 就可将控制器投入运行. 即使过程的动态特性有很大变化, 也不需重新整定控制器的参数. 被用户评价为"梦想成真"的控制器(Dream Controller).

博软今年 3 月推出的 CyboCon2.3 版软件包括了单变量、多变量、抗滞后、解耦、前馈、pH 值、带约束等各种无模型自适应控制器. 它们能方便地解决工业领域中常见的非线性、时变、大滞后、严重耦合、变结构、约束条件苛刻、pH 值控制等等复杂控制问题. 据悉, CyboCon 软件在中国的石化、钢铁、电力、化工、制药等行业恺已获得成功应用, 得到了用户的极高评价.

霍立群