第18卷 第10期
2008年10月

计算机技术与发展
COMPUTER TECHNOLOGY AND DEVELOPMENT

Vol.18 No.10
Oct. 2008

一种基于并行策略的BP改进算法

郭彦伟,王洪国,王 鑫,于 惠
(山东师范大学 信息科学与工程学院,山东 济南 250014)

摘 要:介绍了BP神经网络的基本结构及原理,分析了其收敛慢的原因。为加快其收敛速度,结合带动量梯度下降法提出一种新的算法(PBBP),用多个学习速率不同但结构相同的网络进行并行训练,在每次迭代后都根据误差找出处于最佳状态的网络,并使其它网络的训练参数作适当变化再进行下一次迭代,直到整个网络的误差减小到允许范围内或达到训练次数要求,加快了其收敛速度,能够很好地脱离平坦区。通过在Matlab里编程进行仿真实验证明,该算法是可行的。

关键词:神经网络; BP算法;并行
中图分类号:TP301.6 **文献标识码**:A **文章编号**:1673-629X(2008)10-0110-03

An Improved BP Algorithm Based on Parallel

GUO Yan-wei, WANG Hong-guo, WANG Xin, YU Hui
(School of Information Science and Engineering, Shandong Normal University, Jinan 250014, China)

Abstract: Analysed why the momentum BP algorithm is slow in convergence, describes a new method for the improvement of BP algorithm, named PBBP(parallel based BP algorithm). A method using many networks which are different in learning speed to train by parallel. Then speed up learning process and get out of flat area. The simulation experiments by programing in Matlab show that the improved algorithm is feasible.

Key words: neural network; BP algorithm; parallel

0 引 言

BP算法[1,2]是目前比较流行的神经网络学习算法,是一种典型的误差修正方法,其实质是求均方误差的最小值问题。虽然理论上BP网络能逼近任意的非线性函数,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,训练参数在整个训练过程中无法根据误差的变化进行相应调整,使得网络在学习过程中出现了迭代次数多、收敛速度慢等缺陷。针对这些不足,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[3,4],并且取得了很大进展。

文中通过对带动量的梯度下降法进行改进,提出一种新的算法PBBP(Parallel Based BP Algorithm),用多个不同学习速率的网络进行并行训练,从而减少迭代次数,加快收敛速度。用Matlab进行编程实验,实验结果表明该算法是可行的。

收稿日期:2008-01-17
基金项目:山东省自然科学基金(Q2006G03)
作者简介:郭彦伟(1984-),男,山东聊城人,硕士研究生,研究方向为BP神经网络、最优化理论;王洪国,教授,博士后,研究方向为组合优化算法、数据挖掘、电子政务。

1 BP算法

1.1 算法思想

BP神经网络由输入层、输出层以及处于输入输出层之间的隐含层组成。隐含层又包括单层或多层。对$L(L>3)$层网络,有$L-2$个隐含层。设第$k(k>1)$层的第j个神经元的输入总和为I_j^k,输出为O_j^k,$k-1$层的第i个神经元与k层的第j个神经元的权连接为$W_{ij}^{k-1,k}$,神经元的输出函数为:$f(x)=\frac{1}{1+e^{-x}}$,则:

$$I_j^k=\sum_i W_{ij}^{k-1,k}O_i^{k-1} \tag{1}$$

$$O_j^k=f(I_j^k) \tag{2}$$

$$f'(I_j^k)=O_j^k(I-O_j^k) \tag{3}$$

对于网络的输出层来说,定义它的第m个神经元的输出O_m^k与期望输出d_m的误差E_m有如下形式:$E_m=\frac{1}{2}(O_m^k-d_m)^2$。网络的总误差为

$$E=\sum_m E_m=\frac{1}{2}\sum_m(O_m^k-d_m)^2 \tag{4}$$

由于BP算法采用的是梯度下降法,使权值沿误差函数的负方向变化,则权值修正量为:$\Delta W_{ij}^{k-1,k}=-\eta\cdot\frac{\partial E}{\partial W_{ij}^{k-1,k}}=-\eta\delta_j^k O_i^{k-1}$($\eta>0$,为学习系数,可调节误差

变化的快慢)。

其中，若 k 为输出层，则 $\delta_j^k=(O_j^k-d_m)f'(I_j^k)$；若 k 为中间层 $\delta_j^k=(\sum_j\delta_j^{k+1}W_{ij}^{k,k+1})f'(I_j^k)$。

推导过程请参考文献[5]。

由上可知BP网络学习过程分为两个阶段：一是由前向后正向计算各隐层和输出层的输出；二是由后向前误差反向传播，用于权值修正。通过多次迭代，直到找到全局最优解。

1.2 BP算法收敛速度慢的原因

(1) BP算法中网络参数每次调节的幅度均以一个与网络误差函数或其对权值导数大小成正比的固定因子 η 进行。这样，在误差曲面较平坦处，由于这一偏导数值较小，因而权值参数的调节幅度也较小，需要经过多次调整才能将误差函数曲面降低；而在误差曲面较高曲率处，偏导数较大，权值参数调节的幅度也较大，以致在误差函数最小点附近发生过冲现象，使权值调节路径变为锯齿形，难以收敛到最小点。

(2) BP算法中权值参数的调节是沿误差函数梯度下降方向进行的，但由于网络误差函数矩阵的严重病态性，使这一梯度最速下降方向偏离面向误差曲面的最小点方向，从而急剧加长了权值参数到最小点的搜索路径，大大增加了BP算法的学习时间，这也造成了BP算法收敛速度减慢。

2 带动量的梯度下降算法

基于BP算法的神经元网络，在学习过程中，只需要改变权重，而权重和权重误差导数成正比的。学习系数 η 是学习过程的速率，它是一个常数。η 值越大，权重改变越大。若能选择合适的速率，使它的值尽可能的大但又不至于引起振荡，这样就可以为系统提供一个最快的学习。增大学习效率而又不导致振荡的方法，就是修改反向传播中的学习速率，使它包含一个动量项，具体说，就是每个加权调节量上一项正比例于前次加权变化量的值(即本次权重的修改表达式中引入前次加权的权重修改)。这就要求每次调节完成后，要把该调节量记住，以便在下面的加权调节中使用。设 $\Delta W_{ij}^{k-1,k}(t)$ 为第 t 次迭代的权重变化量，则带动量项的加权调节公式变为：

$$\Delta W_{ij}^{k-1,k}(t+1)=-\eta\delta_j^kO_i^{k-i}+\alpha\Delta W_{ij}^{k-1,k}(t) \quad (5)$$

其中 α 为动量系数，其值可通过实验选取，一般在0.9左右。

加入动量项法后，若相邻两次权值修正的梯度方向是一致的，则可使权值的调整量增大，从而加速收敛；若相邻两次权值修正的梯度方向相反，可使权值的调整量减小，避免了来回振荡，同样可加快收敛速度。实验证明，在用带动量的梯度下降法进行网络的训练时，其收敛速度明显优于标准BP算法。

3 基于并行策略的BP算法

通过对带动量梯度下降法[6]的网络训练过程进行分析，发现其很容易产生平坦区，在平坦区内误差改变很小或不发生变化，要经过很多次的迭代后才能脱离或直到训练结束误差一直停留在平坦区，严重影响网络的收敛速度。带动量梯度下降法通过加动量项改变网络的学习速率，训练期间 η 值不发生变化，而 η 值一般是根据实验或经验来确定，还没有理论指导，所以可能因为 η 的取值不当而达不到网络的最佳迭代次数。

基于并行策略的BP算法是建立 $n(n>1)$ 个结构相同的网络(各网络的学习速率即 η 值不同)，对它们同时进行训练，比较每次迭代后各个网络的误差，以误差下降较快的网络作为起始点进行下一次迭代，直到训练结束。设有两个网络 A_1、A_2，其学习速率分别为 η_1、η_2($\eta_1>\eta_2$)。开始训练时，由于 A_1 学习速率较大，其误差下降明显；当训练接近结束时，A_2 的学习速率较小，可以提高收敛精度。这样，相当于在两个不同学习速率之间根据误差进行动态转换，可以同时发挥两个学习速率的优势。参与训练的网络个数越多，由学习速率带来的优势越大。当某一个网络进入平坦区时，由于其它网络的学习速率不同，可使整个网络自动快速地脱离平坦区；若整个网络同时进入平坦区，由于学习速率不同，大部分情况下也可使整个网络快速脱离，但是当神经元的输出值接近于0或1时，有可能导致权值基本不发生变化从而连续多次使整个网络的误差变化同时为零，这时可改变一次神经元的输出，使下次迭代的权值发生变化。

算法流程如下：

(1) 建立 $n(n>1)$ 个相同的网络(学习速率分别为 $\eta_1,\eta_2,\eta_3,\cdots,\eta_n,\eta_1>\eta_2>\eta_3\cdots>\eta_n$)，并随机初始化权值。

(2) 给定输入 x 和目标输出 d；令 $i=0,k=1$。

(3) 分别计算 n 个网络在第 k 次迭代后的实际输出 $O_1^k,O_2^k,O_3^k,\cdots,O_n^k$，并通过式(4)求得总误差 E_1^k，$E_2^k,E_3^k,\cdots,E_n^k$。若 $E_1^k=E_2^k$ 且 $E_1^{k-1}=E_2^{k-1}$(当 $k=1$ 时，令 $E_1^0=E_2^0$)，则 $i=i+1$；令 $r=\min(E_1^k,E_2^k,E_3^k,\cdots,E_n^k)$，若 r 达到误差精度或网络达到训练次数要求，转到步骤(6)。

(4) 若 $i>3$，则令 $i=0,s=\dfrac{|\lg E_1^k|}{|\lg E_1^k|+1},O_1^k=$

$\max(O_1^k * s,(1-O_1^k) * s)$, $O_2^k = O_1^k, O_3^k = O_1^k, \cdots, O_n^k = O_1^k$;设 E_i^k 为第 k 次迭代后各网络的最小误差,令 $W_1^k = W_i^k, \Delta W_1^k = \Delta W_i^k, \cdots, W_{i-1}^k = W_i^k, \Delta W_{i-1}^k = \Delta W_i^k, W_{i+1}^k = W_i^k, \Delta W_{i+1}^k = \Delta W_i^k, \cdots, W_n^k = W_i^k, \Delta W_n^k = \Delta W_i^k$($W_i^k$ 表示第 k 次迭代后网络 i 的权值), $O_1^k = O_i^k, \cdots, O_{i-1}^k = O_i^k, O_{i+1}^k = O_i^k, \cdots, O_n^k = O_i^k$。

(5)通过如下公式分别对各网络进行权值调整,并记下权值调整量 ΔW_i^{k+1}:

$$\Delta W_1^{k+1} = -\eta_1 \delta O_1^k + \alpha \Delta W_1^k$$

$$\Delta W_2^{k+1} = -\eta_2 \delta O_2^k + \alpha \Delta W_2^k$$

……

$$\Delta W_n^{k+1} = -\eta_n \delta O_n^k + \alpha \Delta W_n^k$$

令 $k = k + 1$,返回步骤(3)。

(6)若 E_i 为到目前为止网络的最小误差,则以网络 i 作为训练完成的网络,删除其它网络。训练结束。

4 仿真实验与结果

通过在 Matlab 中编程对本算法进行了测试。异或问题是典型的线性不可分问题,常用于网络算法的验证,因此文中用它进行仿真实验。构建三个相同结构的三层网络,输入层、隐含层、输出层神经单元个数分别为 2、2、1。输入为{0,0}、{0,1}、{1,0}、{1,1},对应的目标输出为 {0}、{1}、{1}、{0}。随机对权值进行初始化,网络的学习速率分别为 0.16、0.12、0.05,动量系数 α 取 0.91,误差精度为 1e-06,最大迭代次数为 500 次。由于权值为随机选取,因此对网络进行了多次训练,求得平均迭代数,实验结果如表 1 所示。

用带动量 BP 算法进行网络训练,当 η 取 0.2 及 0.12 时,误差曲面经常产生振荡且容易产生平坦区;当 η 取 0.05 时,收敛速度明显减慢。用基于并行策略的 BP 算法(PBBP)进行训练,η_1、η_2、η_3 分别取 0.2、0.12、0.05,误差曲面趋于平滑,很少有平坦区出现,收敛速度很快。结果表明,基于并行策略的 BP 算法能够有效加快网络的收敛速度,避免平坦区引起的迭代次数增加。

表 1 算法执行结果的对比

算法	η 值	平均迭代次数	误差
基于并行策略 BP 算法	(0.2,0.12,0.05)	33	4.77531e-07
带动量 BP 算法	0.2	154	6.36393e-07
带动量 BP 算法	0.12	110	9.51818e-07
带动量 BP 算法	0.05	207	7.60022e-07

5 结束语

基于并行策略的 BP 算法通过对多个采用不同学习速率的网络同时训练,在训练过程中取长补短,充分发挥了学习速率变化对加快网络学习速度的作用,但是也有一定的缺陷,即不同的问题采用多少个网络进行训练合适及学习速率的取值都没有理论依据,只能通过实验来确定。

参考文献:

[1] Rumelhart D E, Hinton G E, Williams R J. Learning representations by back - propagating errors[J]. Nature, 1986, 323(9):533-536.

[2] Rumelhart D E, McClelland J L. Parallel Distributed processing, Explorations in the Microstructure of Cognition, Vol. 1: Foundations[R]. Cambridge, MA: MIT press, 1986.

[3] Jacobs R A. Increased rates of convergence through learning rate adaptation[J]. Neural Networks, 1988(1):295-307.

[4] 黄德双. 神经网络模式识别系统理论[M]. 北京:电子工业出版社,1996.

[5] 刘希玉,刘 宏. 人工神经网络与微粒群优化[M]. 北京:北京邮电大学出版社,2006.

[6] 周志华,曹存根. 神经网络及其应用[M]. 北京:清华大学出版社,2004.

(上接第 109 页)

index for semi structured database[C]// The 27th VLDB Conf. Roma: Morgan Kaufmann, 2001:341-350.

[13] Zou Q, Liu S, Chu W. Ctree: a cmpact tree for indexing XML data[C]// the 6th ACM WI DM 04. New York: [s. n.], 2004:39-46.

[14] XML Schema 1.1 specification[EB/OL]. 2000-04. http://www.w3.org/XML/Schema.

[15] Java DTD Parser[CP/OL]. 2004-07. http://www.wutka.com/dtdparser.html.

[16] Wang G, Liu M. Extending XML Schema with Nonmonotonic Inheritance[C]// In Proceedings of 1st International Workshop on XML Schema and Data Management (ER Workshop XSDM'03). Chicago, Illinois, USA: [s. n.], 2003.

[17] 张晓琳,王国仁. 用继承扩展 XML-RL[J]. 小型微型计算机系统,2005,26(2):243-247.

[18] 张晓琳,谭跃生,周 健. 用继承扩展 XML Schema[J]. 计算机工程与应用,2006,42(4):179-182.

[19] Document Object Model (DOM Level 3 Core)[EB/OL]. 2002-04. http://www.w3.org/DOM/.

[20] 张晓琳,王国仁. 面向对象的 XML 数据的存储模式研究[J]. 小型微型计算机系统,2004,25:11-13.