新型样条权函数神经网络的云计算研究

张代远
(南京邮电大学 计算机学院,江苏 南京 210003)

摘 要:采用权函数训练神经网络是近些年发展起来的一种算法。该算法有许多优点,例如可以直接求得全局最优点,有很好的泛化能力,训练后的权函数能够反映隐含在样本内部的有价值的信息特征等。因此进一步提高算法效率就显得十分重要。为了进一步提高运算速度,文中将神经网络与云计算相结合,采用云计算服务对一种新型的三次样条权函数神经网络算法的性能进行了分析,提出了云计算的定义,研究了三次样条权函数神经网络算法的并行机制。结果表明,采用云计算能够大幅提高三次样条权函数神经网络算法的效率。

关键词:云计算;人工智能;前馈神经网络;三次样条函数;权函数;全局最小;插值

中图分类号:TP18;TP183　　**文献标识码**:A　　**文章编号**:1673-629X(2013)07-0057-05

doi:10.3969/j. issn. 1673-629X. 2013. 07. 014

Research on Cloud Computing for Neural Network of a New Kind of Spline Weight Functions

ZHANG Dai-yuan
(College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: Training neural network using weight functions is a new kind of algorithm developed in recent years, which has many advantanges, such as finding globe minima directly, good performance of generalization, extracting some useful information inherent in the problems and so on. It is very important to improve the efficiency for this new kind of algorithm. To improve the training speed, combine the neural networks with cloud computing, and analyze the performance of cloud computing services for the algorithm of training neural networks by cubic spline weight functions (NNCSWFs). Attempt to give the definition for cloud computation, and study the parallel mechanism to improve the computing performance. The results indicate that, by using cloud computing services, the work efficiency for NNCSWFs algorithm is much higher than before.

Key words: cloud computing; artificial intelligence; feedforward neural network; cubic spline function; weight function; global minima; interpolation

0 引 言

文中研究一种新型样条权函数神经网络的云计算理论与方法。

以下的几个概念有助于文中的讨论。神经网络各个神经元的变换函数的集合,称为该神经网络的网络函数。网络各层间的连接权值和阈值称为连接参数。网络各层神经元的个数称为结构参数。网络的连接参数和结构参数统称为网络参数。网络的神经元之间的连接方式称为网络的拓扑结构。

许多学者为了改进传统算法(例如BP算法或RBF算法)进行了大量研究。比如,一些逼近方法的研究[1,2],提高训练速度[3,4],采用剪裁方法改变动态网络的结构[5,6],优化初始权值或者其他参数[7],训练过程的在线收敛性[8],使用附加的参数确定收敛性,在线梯度方法[9],采用遗传算法或者演化计算方法调整网络的参数[10],以及一些混合方法等等。不幸的是,以上的这些改进没有从根本上克服梯度下降类算法的本质缺点。另一方面,这些改进工作基本上集中在加快收敛速度、提高运算精度、选择合适的初值、改善收敛性等方面,却没有关注对存储在网络中的权值的物理意义的解释上。

对于传统的前馈神经网络,在训练后网络的权值是常数。这些常数权可以按照一定的排列方式组成一

收稿日期:2012-10-12　　**修回日期**:2013-01-20　　**网络出版时间**:2013-04-08

基金项目:江苏高校优势学科建设工程资助项目(yx002001)

作者简介:张代远(1957-),男,教授,博士,研究生导师,研究方向为人工智能、计算机体系结构、计算机应用等。

网络出版地址:http://www.cnki.net/kcms/detail/61.1450.TP.20130408.1600.029.html

万方数据

个权矩阵,但是,由于矩阵中的权仅仅是一些离散的数据,很难在这些权值与训练过的样本之间建立起内在的对应关系,也就是说,人们很难从存储在网络的权值中提取有用的规则。由于常数权值很难反映出训练样本的特征,因此训练好的网络通常是一个非线性输入输出系统,这些求得的权值通常无法用来获取隐含在系统内部的、有价值的规则。

文献[11]提出了一种能够求得全局最优(即全局最小)解的算法(称为代数算法),代数算法克服了 BP 算法的缺点,速度快、精度高是其主要优点。该算法也仅仅训练网络的连接参数,可以方便确定网络的结构参数。代数算法的主要缺点是:为了求得全局最优点,要求隐层神经元个数等于样本个数。

文献[11]提出了一种新型学习算法,克服了已有方法的上述缺点。这种新型权函数神经网络算法能够克服传统神经网络训练的权值难以反映训练样本信息的缺陷,将训练后的常数权改造成权函数,并希望能够反映隐含在样本内部的有价值的的信息特征。

云计算的基本概念可以追溯到 20 世纪 60 年代,当时人们就提出计算是一种服务的概念,但是受到计算机技术和网络技术的限制,并没有引起人们更多的关注。但随着计算机技术和网络技术的进步,特别是对大规模计算的需求,人们对云计算日益关注。云计算的性能如何,也是人们研究的热点之一。例如:文献[12]分析了在科学计算中云计算服务的性能。文献[13]研究了云计算环境下,使用分布负载机制,对大规模多项式乘积计算总时间开销的最优化方法。文献[14]研究了一种近似分析方法,来估计云服务的性能。文献[15]研究了一种随机编程模式的优化云资源提供算法。

文中将重要的智能计算方法—神经网络与云计算相结合,研究了采用云计算技术的神经网络方法。由于样条权函数神经网络具有很多优点[11],文中对文献[11]进行了进一步的深入研究,给出了文献[11]算法的并行计算原理,并引入了云计算的定义,采用云计算思想,实现了文献[11]算法的云计算原理,分析了算法的时间效率。

1 网络拓扑结构

在图 1 中,z_{sj}、z_j 分别表示输出层第 j 个神经元的实际输出、输出层第 j 个神经元的目标值。

由于只有 1 层权需要训练,不失一般性,可以只考虑其中一个神经元的连接权。图 1 中标有 $\mathrm{Add}_j(j=1,2,\cdots,n)$ 的圆圈为第 j 个神经元。这里仅仅研究与第 j 个神经元 Add_j 相连接的权函数的确定方法,其余神经元的权函数的确定完全类似。

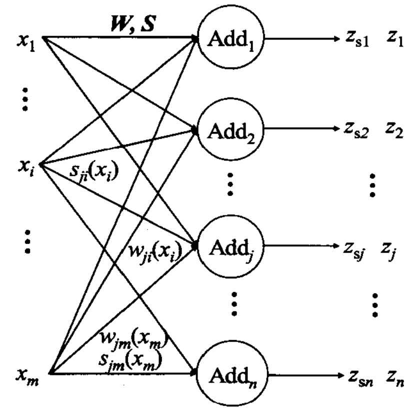

图 1 神经网络结构

假设每一个输入、输出样本都是由 m、n 维向量构成(见图1),总共有 $N+2$ 个需要训练的样本,$s_{ji}(x_i)$ 表示神经元 j 与第 i 个输入节点相连的权,x_i 表示 m 维输入向量的第 i 个分量。由于有 $N+2$ 个需要训练的样本,因此节点 x_i 将有 $N+2$ 个输入量。将这 $N+2$ 个输入量按照输入样本的顺序组成一个 $N+2$ 维向量,记为:

$$\boldsymbol{x}_i=(x_{i0},x_{i1},\cdots,x_{i(N+1)}) \quad (1)$$

同样,也将这 $N+2$ 个输入量所对应的目标向量组成一个 $N+2$ 维向量,记为:

$$\boldsymbol{z}_j=(z_{j0},z_{j1},\cdots,z_{j(N+1)}) \quad (2)$$

(1)、(2) 式中的 x_{ip}、z_{jp} 分别代表输入层第 i 个输入节点的第 $p(p=0,1,\cdots,N+1)$ 个输入样本和输出层第 j 个输出节点的第 p 个目标值,是一个 1 维变量(标量)。另一方面,将 $s_{ji}(x_i)$ 看成一元函数,自变量 x_i 取 (1) 式中的 $N+2$ 个数值。现在的问题是要确定函数 $s_{ji}(x_i)$。现在已知的是自变量的取值,如果能够确定 $s_{ji}(x_i)$ 的函数值(或输出值),就可以通过插值的方法确定函数 $s_{ji}(x_i)$。

将(2) 式的第 j 个节点的目标向量再分解为 m 个向量。这样分解的目的就是将目标向量进行适当的分配,使得能够对应于输出层的第 j 个节点相连接的 m 个权。一种最简单的分解方法就是令

$$\boldsymbol{z}_j=\eta_{j1}\boldsymbol{z}_j+\eta_{j2}\boldsymbol{z}_j+\cdots+\eta_{jm}\boldsymbol{z}_j \quad (3)$$

其中

$$\sum_{i=1}^{m}\eta_{ji}=1,\eta_{ji}\geqslant 0 \quad (4)$$

按照(3) 式的分配方法,权函数 $s_{ji}(x_i)$ 的输入量由(1) 式决定,输出量则为 $\eta_{ji}(z_{j0},z_{j1},\cdots,z_{j(N+1)})$。对应的插值点为:

$$(x_{ip},\eta_{ji}z_{jp}),p=0,1,\cdots,N+1 \quad (5)$$

由(5) 式,就可以根据插值理论确定权函数 $s_{ji}(x_i)$。这里采用三次样条函数来确定权函数 $s_{ji}(x_i)$。

神经元 Add_j 是加法器,其功能是将所有与其相连

万方数据

的输入节点的输入量求和后直接输出。

由图1可见,网络拓扑结构非常简单。只需训练1层权函数(传统算法常常需要训练多层权值)。网络训练所需的神经元个数 n 与样本个数无关。需要训练的权函数的个数可以简单地表示成神经网络的输入、输出层的节点个数之积 mn。

2 算法原理以及并行机制的引进

首先简化符号。假设需要训练的样本点有 $N+2$ 个,即 $\{x_p,y_p\}_{p=0}^{N+1}$,并且设 $a=x_0<x_1<\cdots<x_{N+1}=b$。三次样条函数由分段三次多项式决定(见(6)式)。

$$s(x)=s_p(x)=\sum_{k=0}^{3}A_{pk}(x-x_p)^k \tag{6}$$

(6)式中 $x\in[x_p,x_{p+1}]$,$p=0,1,\cdots,N$。(6)式中各个分段多项式要满足以下关系:

$$\begin{cases}s_{p-1}^{(a)}(x_p)=s_p^{(a)}(x_p) & a=0,1,2\\ s_p(x_p)=y_p\\ s_0(x_0)=y_0\\ s_N(x_{N+1})=y_{N+1}\end{cases} \tag{7}$$

(7)式中 $p=1,\cdots,N$。根据(7)式知道,(6)式的二阶导数是线性分段函数,于是有:

$$s_p''(x)=s''(x_p)\frac{x-x_{p+1}}{x_p-x_{p+1}}+s''(x_{p+1})\frac{x-x_p}{x_{p+1}-x_p} \tag{8}$$

将(8)式积分两次,并利用(7)式的前两个表达式得到:

$$s_p(x)=-\frac{M_p}{6h_p}(x_{p+1}-x)^3+\frac{M_{p+1}}{6h_p}(x-x_p)^3+\left(\frac{y_p}{h_p}-\frac{M_ph_p}{6}\right)(x_{p+1}-x)+\left(\frac{y_{p+1}}{h_p}-\frac{M_{p+1}h_p}{6}\right)(x-x_p) \tag{9}$$

现在的问题是求得(9)式的 M_p,为此,计算(9)式的一阶导数得:

$$s'_p(x)=-\frac{M_p}{2h_p}(x_{p+1}-x)^2+\frac{M_{p+1}}{2h_p}(x-x_p)^2-\left(\frac{y_p}{h_p}-\frac{M_ph_p}{6}\right)+\left(\frac{y_{p+1}}{h_p}-\frac{M_{p+1}h_p}{6}\right) \tag{10}$$

将 p 写成 $p-1$ 得:

$$s'_{p-1}(x)=-\frac{M_{p-1}}{2h_{p-1}}(x_p-x)^2+\frac{M_p}{2h_{p-1}}(x-x_{p-1})^2-\left(\frac{y_{p-1}}{h_{p-1}}-\frac{M_{p-1}h_{p-1}}{6}\right)+\left(\frac{y_p}{h_{p-1}}-\frac{M_ph_{p-1}}{6}\right) \tag{11}$$

于是有

$$s'_p(x_p)=-\frac{M_ph_p}{3}-\frac{M_{p+1}h_p}{6}+d_p \tag{12}$$

$$s'_{p-1}(x_p)=\frac{M_ph_{p-1}}{3}+\frac{M_{p-1}h_{p-1}}{6}+d_{p-1} \tag{13}$$

(12)、(13)式中的 $h_p=x_{p+1}-x_p$,$d_p=(y_{p+1}-y_p)/h_p$,$h_{p-1}=x_p-x_{p-1}$,$d_{p-1}=(y_p-y_{p-1})/h_{p-1}$。利用(12)、(13)式,根据一阶导数的连续性以及边界条件得到:

$$\begin{cases}g_0M_0+g_1M_1=u_0\\ h_{p-1}M_{p-1}+2(h_{p-1}+h_p)M_p+h_pM_{p+1}=u_p\\ g_NM_N+g_{N+1}M_{N+1}=u_{N+1}\end{cases} \tag{14}$$

其中,$p=1,2,\cdots,N$,$u_p=6(d_p-d_{p-1})$,g_0、g_1、u_0、g_N、g_{N+1}、u_{N+1} 是与边界条件相关的常数。

假设端点处邻近两点是线性关系,常用的第一、第二类边界条件是其特例。

假设 y'_0、y'_{N+1} 是端点处已知的一阶导数值,根据(10)、(11)式,利用连续性条件 $s'_p(x_p+0)\mid_{p=0}=y'_0$、$s'_{p-1}(x_p-0)\mid_{p=N+1}=y'_{N+1}$,可求得第一类边界条件为:

$$\begin{cases}2h_0M_0+h_0M_1=6(d_0-y'_0)\\ h_NM_N+2h_NM_{N+1}=6(y'_{N+1}-d_N)\end{cases} \tag{15}$$

(15)式相当于(14)式中的 $g_0=2h_0$、$g_1=h_0$、$u_0=6(d_0-y'_0)$、$g_N=h_N$、$g_{N+1}=2h_N$、$u_{N+1}=6(y'_{N+1}-d_N)$。

第二类边界条件更加简单,为:

$$\begin{cases}M_0=y''_0\\ M_{N+1}=y''_{N+1}\end{cases} \tag{16}$$

(16)式中 y''_0、y''_{N+1} 是端点处已知的二阶导数值,对应于(14)式中的 $g_0=1$、$g_1=0$、$u_0=y_0$、$g_N=0$、$g_{N+1}=1$、$u_{N+1}=y''_{N+1}$。

对于第一类、第二类边界条件,(14)式的系数矩阵严格对角占优,根据矩阵理论知道(14)式有唯一解。将(14)式求得的 M_p 带入(11)式,就可以求得相应的样条函数表达式。

将 x 改写成 x_i,M_p 改写成 M_{jip},h_p 改写成 h_{ip},由(3)、(5)式,(9)式可以写成:

$$s_{jip}(x)=-\frac{M_{jip}}{6h_{ip}}(x_{i(p+1)}-x_i)^3+\frac{M_{ji(p+1)}}{6h_{ip}}(x_i-x_{ip})^3+\left[\frac{\eta_{ji}z_{jp}}{h_{ip}}-\frac{M_{jip}h_{ip}}{6}\right](x_{i(p+1)}-x_i)+\left[\frac{\eta_{ji}z_{j(p+1)}}{h_{ip}}-\frac{M_{ji(p+1)}h_{ip}}{6}\right](x_i-x_{ip}) \tag{17}$$

其中 $p=0,1,\cdots,N$。将子区间的函数 $s_{jip}(x_i)$ 拼接起来就得到整个区间上的样条函数,记为 $S_{ji}(x_i)$。

权函数 $S_{ji}(x_i)$ 为三次样条函数,它是输入样本 x_i 的一元函数,不是常数。权函数在一定的意义上可以反映隐含在样本内部的重要信息。这是传统方法得到的常数权所无法获得的。

显然,只要样本给定,不同的输入输出节点上的样条权函数 $S_{ji}(x_i)$ 彼此无关,完全可以并行计算。算法需要求解彼此独立的三次样条权函数 $S_{ji}(x_i)$,它们都是自变量 x_i 的一元函数,其个数为:

$$N_W=mn \tag{18}$$

万方数据

这种并行机制的引入有助于采用云计算方法来提高神经网络的训练效率。

3 泛化能力分析

样条权函数神经网络的另外一个优点是可以方便进行误差分析,并且有很好的泛化能力,结果如下:

$$\begin{aligned} J &= \max_j J_j = \max_j \| z_j^{(a)} - z_{sj}^{(a)} \|_\infty \\ &= \max_j \| \sum_{i=1}^{m} w_{ji}^{(a)}(x_i) - \sum_{i=1}^{m} s_{ji}^{(a)}(x_i) \|_\infty \\ &\leqslant \max_j (\sum_{i=1}^{m} \| w_{ji}^{(a)}(x_i) - s_{ji}^{(a)}(x_i) \|_\infty) \\ &\leqslant m \cdot \max_j (\max_i (\| w_{ji}^{(a)}(x_i) - s_{ji}^{(a)}(x_i) \|_\infty)) \\ &\leqslant m \cdot \max_j (\max_i (B_{ia} \cdot h_i^{4-\alpha} \cdot \| w_{ji}^{(4)}(x_i) \|_\infty)) \end{aligned} \tag{19}$$

其中 $h_i = \max\{h_{i0}, h_{i1}, \cdots, h_{iN}\}$,是相邻两个样本插值点之间的距离。

4 样条权函数神经网络的云计算实现原理

要理解云计算,就要先定义“云”。云计算目前没有统一的定义。文中作者给出以下一些定义。

定义 1:云是一种集合,该集合中的各个元素可以具有不同的类型和属性。

云中的元素也称为云节点。云中的元素可以有软件,也可以有硬件。例如,可以将网络中大量各种不同类型的存储设备通过各种应用软件集合起来协同工作,共同对外提供数据存储、计算和业务访问服务。

定义 2:云的部分元素构成的子集称为子云。

定义 3:云中的一些元素,这些元素是具有一定功能的硬件实体,称为硬件云。

定义 4:云中一些元素,这些元素能够利用硬件云并实现一定功能的软件实体称为软件云。

显然,硬件云和软件云都是子云。

定义 5:向云输入用户数据(需求)信息,通过云完成相应的计算,并将计算结果反馈给用户的过程称为云计算。

因此云计算本质上仍然是一种信息变换过程。云计算的一个重要特征就是将用户的任务分解成子任务,通过云中的各个不同元素分别执行,以提高效率。

为了使用云计算实现样条权函数神经网络的训练,重要的前提是能够将样条权函数神经网络算法分解成子任务,这一点是可以做到的。这是因为每个权函数是相互独立的,只需要将(14)式分配给云中不同的云节点(元素),就能够提高时间效率,具体分析如下。

若分配给第 i 个元素的任务数为 λ_i,λ_i 是整数,Δt_i 是第 i 个云节点元素的全速执行时间,称为云节点的执行开销,Δu_i 是第 i 个元素的加载、存储、传输、操作系统等的时间开销之和,称为云节点的系统开销。假设每一个云节点运行多个同类任务的时候,花费在每一个任务上的执行开销都相等,为 Δt_i,花费在每一个任务上的系统开销都相等,为 Δu_i,则任务数 λ_i 的分配原则为:

$$\begin{cases} T_C = \min \max_I \{ \lambda_i(\Delta t_i + \Delta u_i) \mid i = 1,2,\cdots,N_C \} \\ \text{s.t.} \sum_{i=1}^{N_C} \lambda_i = mn \end{cases} \tag{20}$$

假设只有一个云节点,则有:

$$T_1 = mn(\Delta t_1 + \Delta u_1) \tag{21}$$

于是,加速比为:

$$S = \frac{T_1}{T_C} = \frac{mn(\Delta t_1 + \Delta u_1)}{\min \max_i \{ \lambda_i(\Delta t_i + \Delta u_i) \mid i = 1,2,\cdots,N_C \}} \tag{22}$$

若假设每个云节点的性能都相同,即执行开销 Δt 和系统开销 Δu 都相等,则(20)式可以写成

$$T_C = \left[\frac{mn}{N_C}\right] (\Delta t + \Delta u) \tag{23}$$

此时加速比为:

$$S = \frac{T_1}{T_C} \approx N_C \tag{24}$$

通常,$N_C \geqslant 1$,于是,(24)式说明样条权函数神经网络训练算法能够按照云计算实现加速,提高效率。而传统算法(BP)通常难以分解为彼此独立的子任务,所以难以采用云计算实现加速。

5 数值仿真实验

例 1:本实验说明样条权函数神经网络能够反映隐含在样本内部的信息特征。样本由以下公式得到:

$$\begin{cases} z_1 = (x_1^2 + x_2^2 + x_3^3) \\ z_2 = e^{-(x_1+x_2+x_3)} \cos(x_1 + x_2 + x_3) \end{cases} \tag{25}$$

学习曲线为:

$$x_1 = t, x_2 = 0.5t, x_3 = 2t + 1 \tag{26}$$

其中 $t \in [0,3]$。这 30 个样本点等距分布,由(25)式得到。参数 t 的取值为:

$$t = 0 + (3 - 0)p/(30 - 1) = 3p/29 \tag{27}$$

这里 $p = 0,1,2,\cdots,29$。分配系数为 $\eta_{ji} = 1/3$。

目标输出由(25)和(26)式计算,见图 2。其中一个样条权函数见图 3,可看出较好地反映了样本特征。

例 2:本实验考察引入云计算后系统的时间效率。为了既简单又能说明问题,假设云中的每个节点都是单处理器计算机系统,而且性能都相同。实验结果见表 1。从表 1 可以明显看出,采用云计算比使用

万方数据

单处理器计算机系统的时间效率有大幅的提高。应该指出的是,之所以能采用云计算大幅提高计算效率是因为能够将并行机制有效引入样条权函数神经网络之中,这一点传统方法(如BP算法等)是很难做到的。

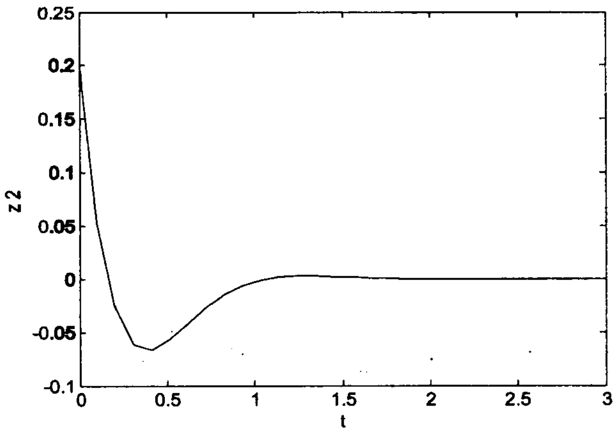

图2　(25)和(26)式求出的 z_2 的理论值

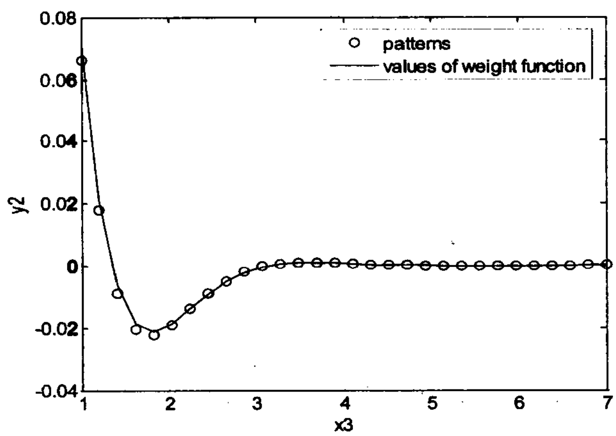

图3　样条权函数 $s_{23}(x_3)$

表1　云计算训练时间 ($p = 1000, N_C = 2048$)

mn	2000	4000	6000	8000	10000
单机消耗(s)	4876.7	9753.3	14630.0	19506.7	24383.3
云计算时间消耗(s)	2.4	4.9	7.3	9.8	12.2

6 结束语

样条权函数神经网络算法可直接求得全局最优点,有很好的泛化能力,训练后的权函数能够反映隐含在样本内部的有价值的信息特征。样条权函数神经网络算法所需要求解的样条权函数彼此相互独立,其个数与样本个数无关,每个三次样条权函数的计算过程是相同的,其时间开销是一样的(见(14)式)。这些重要特点使得能够方便引入并行机制,可采用云计算服务对其进行分析计算。结果指出,采用云计算能够大幅提高三次样条权函数神经网络算法的效率。而传统方法由于网络结构复杂、各个权值耦合紧密、难以独立训练,使得难以并行化,因而难以应用云计算实现加速。

参考文献:

[1] Hou M Z, Han X L. Constructive Approximation to Multivariate Function by Decay RBF Neural Network[J]. IEEE Trans. on Neural Netw. ,2010,21(9):1517-1523.

[2] Wedge D, Ingram D, McLean D, et al. On global-local artificial neural networks for function approximation [J]. IEEE Trans. on Neural Netw. ,2006,17(4):942-952.

[3] Ergezinger S, Tomsen E. An accelerated learning algorithm for multilayer perceptrons: optimization layer by layer[J]. IEEE Trans. on Neural Netw. ,1995,6(1):31-42.

[4] Ampazis N, Perantonis S J. Two highly efficient second-order algorithms for training feedforward networks[J]. IEEE Trans. on Neural Netw. ,2002,13(5):1064-1073.

[5] Huang G B, Saratchandran P, Sundararajan N. A generalized growing and pruning RBF (GGAP-RBF) neural network for function approximation[J]. IEEE Trans. on Neural Netw. , 2005,16(1):57-67.

[6] Bortman M, Aladjem M. A Growing and Pruning Method for Radial Basis Function Networks[J]. IEEE Trans. on Neural Netw. ,2009,20(6):1039-1045.

[7] Behera L, Kumar S, Patnaik A. On Adaptive Learning Rate That Guarantees Convergence in Feedforward Networks[J]. IEEE Trans. on Neural Netw. ,2006,17(5):1116-1125.

[8] Xu Z B, Zhang R, Jing W F. When Does Online BP Training Converge[J]. IEEE Trans. on Neural Netw. ,2009,20(10):1529-1539.

[9] Wu W, Feng G R, Li Z X, et al. Deterministic convergence of an online gradient method for BP neural networks[J]. IEEE Trans. on Neural Netw. ,2005,16(5):533-540.

[10] Khashman A. A Modified Backpropagation Learning Algorithm with Added Emotional Coefficients[J]. IEEE Trans. on Neural Netw. ,2008,19(11):1896-1909.

[11] 张代远. 神经网络新理论与方法[M]. 北京:清华大学出版社,2006.

[12] Iosup A, Ostermann S, Yigitbasi M N, et al. Performance Analysis of Cloud Computing Services for Many-tasks Scientific Computing[J]. IEEE Trans. on Parallel and Distributed Systems,2011,22(6):931-945.

[13] Iyer G N, Veeravalli B, Krishnamoorthy S G. On Handling Large-scale Polynomial Multiplications in Compute Cloud Environments Using Divisible Load Paradigm[J]. IEEE Trans. on Aerospace and Electronic Systems, 2012, 48(1): 820-831.

[14] Khazaei H, Misic J, Misic V B. Performance Analysis of Cloud Computing Centers Using M/G/m/m+r Queueing Systems [J]. IEEE Trans. on Parallel and Distributed Systems,2012, 23(5):936-943.

[15] Chaisiri S, Lee Bu-Sung, Niyato D. Optimization of Resource Provisioning Cost in Cloud Computing[J]. IEEE Trans. on Service Computing,2012,5(2):164-177.

万方数据

万方数据 WANFANG DATA 文献链接

新型样条权函数神经网络的云计算研究

作者： 张代远， ZHANG Dai-yuan
作者单位： 南京邮电大学计算机学院,江苏南京,210003
刊名： 计算机技术与发展ISTIC
英文刊名： Computer Technology and Development
年，卷(期)： 2013,23(7)

本文链接：http://d.g.wanfangdata.com.cn/Periodical_wjfz201307014.aspx