

文章编号: 2095-0411 (2014) 01 - 0052 - 05

改进型 RBF 神经网络在线的学习算法^{*}

张小鸣, 王研顺

(常州大学 信息科学与工程学院, 江苏 常州 213164)

摘要: 为了更好的满足嵌入式应用领域和实时性环境的要求, 在资源分配网络 (RAN) 的基础上提出了一种改进型径向基函数 (RBF) 神经网络在线学习算法。在网络参数调整过程中引入了分级学习率因子, 根据理论输出和网络输出误差绝对值的大小选择不同的学习率因子参与学习过程。在 VC++6.0 编程环境中进行的软件仿真试验表明: 相对于传统的 RBF 神经网络在线学习算法, 改进型 RBF 神经网络在线学习算法在不增加网络规模的情况下可以进一步减小输出误差。

关键词: 径向基函数神经网络; 在线学习算法; 分级学习率

中图分类号: TP 301.6

文献标识码: A

doi: 10.3969/j.issn.2095-0411.2014.01.012

Research on Improved On - Line Learning Algorithm of RBF Neural Network

ZHANG Xiao-ming, WANG Yan-shun

(School of Information Science and Engineering, Changzhou University, Changzhou 213164, China)

Abstract: In order to meet the requirements of embedded applications and real - time environment better, an improved on - line learning algorithm of Radial Basis Function (RBF) neural network based on resource allocation network (RAN) is proposed. A hierarchical learning rate factor is introduced into this algorithm in the network parameter adjustment process. Different learning rate factors are selected to take part in the learning process according to the absolute value of the error between theoretical output and network output. VC++6.0 software simulation experiments show that compared with the traditional RBF neural network online learning algorithm, the improved RBF neural network on - line learning algorithm can further reduce the output error without increasing the size of the network.

Key words: RBF neural network; on - line learning algorithm; hierarchical learning rate

RBF 神经网络学习算法根据参数修正发生的时间可以分为离线学习算法和在线学习算法。离线学习算法是将学习样本集中的所有样本一起送入网络中参加学习, 这就要求在学习开始前就必须准备好所有需要的样本, 而这在很多实际应用中是难以实现的。针对离线学习算法的不足, Platt 在 1991 年提出了 RAN (Resource Allocating Network,

资源分配网) 模型, 它可以根据输入数据的“新颖性”动态的增加网络的隐层神经元个数, 是一种全新的在线学习神经网络算法。但其依然存在学习速度慢、未引入删除策略等诸多缺点。为了进一步提高 RBF 网络在线学习的速度、精度和效率, 研究学者提出了多种改进的方法, 例如采用递推最小二乘法^[1]、敏感度法^[2]、梯度下降法^[3]等方法对网络

* 收稿日期: 2013 - 06 - 24

作者简介: 张小鸣 (1958-), 男, 安徽合肥人, 教授。

参数进行调整; 在改进网络结构方面, 结合粒子群^[4]、蚁群^[5]、遗传进化^[6-8]、模糊逻辑^[9-10]等其他优化算法提高网络优化的准确率; 在提高网络效率方面, 引入隐层节点删除策略避免网络结构的无限膨胀从而降低计算量等^[11-12]。

本文在 RAN 算法的基础上结合文献 [1] 中的方法提出一种改进型 RBF 神经网络在线学习算法, 从参数优化的角度对 RBF 神经网络在线学习算法进行改进。传统学习算法中对隐层节点中心和输出偏移进行调整时采用固定不变的学习率, 导致节点中心和输出偏移不能根据网络学习的需要按需调整, 输出误差仍然较大。本文提出的分级学习率因子是将学习率分为不同的等级, 根据学习过程中网络输出和理论输出的误差绝对值的大小划分出不同的区间, 每个区间都有与之对应的学习率因子, 在每次迭代学习过程中根据误差绝对值的所在区间选择相应的学习率因子参与计算, 从而得到相应的节点中心和输出偏移。软件仿真实验结果表明, 该方法在不增加隐含层节点的情况下可以有效的减小网络输出误差。

1 RBF 神经网络及在线学习算法

1.1 RBF 神经网络结构

RBF 神经网络是 3 层结构的前馈式神经网络, 分别为输入层、隐层和输出层, 基本结构如图 1 所示。输入层节点的作用是将输入样本传递到隐层; 隐层节点由径向基函数构成, 作用是对输入样本进行非线性变换; 输出层节点通常是简单的线性函数, 负责实现网络结果输出。

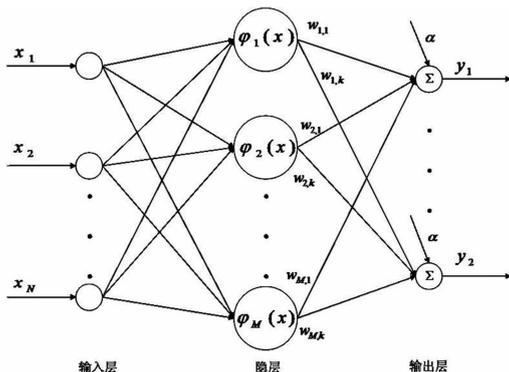


图 1 RBF 神经网络基本结构

Fig. 1 Basic structure of RBF neural network

RBF 神经网络的非线性映射能力是由隐层基函数体现的, 常用的径向基函数是高斯函数, 其基本表达式为:

$$\varphi(r) = \exp\left(-\frac{r^2}{2\sigma^2}\right) \quad \sigma > 0; r \in R \quad (1)$$

对于单输出的 RBF 神经网络, 第 i 个隐层单元的输出为:

$$\varphi_i(x_t) = \exp\left(-\frac{\|x_t - c_i\|^2}{2\sigma_i^2}\right) \quad (2)$$

$i=1, 2, \dots, M; t=1, 2, \dots, N$

式中 $\|\cdot\|$ 为欧几里德范数, x_t 为第 t 个输入样本, c_i 为第 i 个隐层节点的中心值, σ_i 为第 i 个隐层节点的宽度。 M 是隐层节点个数, N 是输入样本个数。

整个网络的输出方程为:

$$f(x_t) = \sum_{i=1}^M \omega_i \varphi_i(x_t) + \alpha \quad (3)$$

式中 ω_i 为输出空间与第 i 个隐层节点的连接权值, α 为输出偏移。

从以上公式可以看出, RBF 神经网络主要包含 4 类参数: 网络隐层节点数及其中心值、隐层节点宽度、输出层线性连接权值、输出偏移。常见的学习算法都是通过对这 4 类参数进行设置和调整来设计构造出适应不同实际需求的 RBF 神经网络。在以往的研究中多采用离线学习法, 即将所有训练样本一次性送入网络中同时参与学习过程, 从而得到最佳的网络结构。这就需要在开始学习前准备好所有的训练样本, 但在很多实际应用中是不可能实现的。因此为了适应实际需要, 人们研究出了 RBF 神经网络在线学习算法。

1.2 RBF 神经网络在线学习算法

RBF 神经网络在线学习算法是将训练样本顺序地逐个地送入网络中参与学习过程, 事先不需要知道样本空间容量, 针对不同的问题可以自适应地确定 RBF 神经网络的隐层节点数及其中心、输出层连接权值及输出偏移等。本文在 RAN 算法的基础上结合文献 [1] 中提出的改进方法, 采用分级学习率的方法, 形成了一种新的改进型 RBF 神经网络在线学习算法。在学习前, 构建一个初始隐层节点数为零的 RBF 神经网络, 开始学习后, 对每一个送入网络的样本判断其是否满足“新颖性”要求, 如果满足条件就增加新的隐层节点, 并选择相应的节点中心、宽度以及连接权值, 如果不满足“新颖性”要求, 则采用梯度下降法和递推最小二乘法对网络参数进行精调, 直到满足误差精度要求或者达到最大学习次数, 最后再通过隐层节点删除策略删除冗余节点。

1.2.1 隐层节点增加规则

根据 RAN 算法规则, 隐层节点的增加是以“新颖性”作为标准的。当输入样本远离现存的基函数中心, 并且其网络输出与实际输出之间存在较大的误差时, 则认为这个样本是“新颖的”。RAN 算法中用于增加隐层节点的“新颖性”条件包含以下两部分内容:

$$\min \| x_t - c_i \| > \delta (t) \tag{4}$$

$$\| y (x_t) - f (x_t) \| > \epsilon_1 \tag{5}$$

式中 x_t 为第 t 次输入的样本, $y (x_t)$ 为网络理论输出, $f (x_t)$ 为网络实际输出, $\delta (t)$ 和 ϵ_1 为阈值, 且:

$$\delta (t) = \max (\delta_{\max} \tau^t, \delta_{\min}) \tag{6}$$

$\delta (t)$ 是在第 t 个样本输入时网络拟合精度的大小。其中 δ_{\max} 和 δ_{\min} 分别为输入样本 x_t 与已存在的中心之间的最大和最小距离, τ 为衰减常数, 满足 $0 < \lambda < 1$ 。除了上述两条“新颖性”标准, 为了减少噪声的影响, 进一步提高网络的泛化能力, 借鉴文献 [1] 引入第 3 条“新颖性”标准, 即通过引入以一定数目的输出误差为基础构成的均方根 (RMS, Root Mean Square) 滑动窗来防止受突发噪声影响严重的输入样本成为隐层节点, 从而减小网络学习误差。其表达式为:

$$e_{\text{rms}} = \sqrt{\frac{\sum_{i=t-(M-1)}^t (y (x_t) - f (x_t))^2}{M}} > \epsilon_2 \tag{7}$$

式中 M 为滑动窗宽度即累计误差个数, ϵ_2 为阈值。只有当式 (4)、(5)、(7) 同时满足时才增加一个新节点, 其参数为:

$$c_{i+1} = x_t \tag{8}$$

$$\omega_{i+1} = y (x_t) - f (x_t) \tag{9}$$

$$\sigma_{i+1} = K \| x_t - c_i \| \tag{10}$$

式中 K 为重叠因子。

1.2.2 冗余节点删除策略

由式 (3) 可知, RBF 网络输出层的输出是对隐层节点输出的线性加权。隐层节点的数量和质量直接影响到整个网络的输出。如果任由隐层节点无限制增加下去, 一方面使计算量大幅度增加, 降低了网络学习速度, 另一方面由于可能存在的低质量节点一直参与网络学习过程会影响网络学习最终结果的准确性。针对这一问题, 本文以隐节点的输出和输出的连接权值对整个网络的贡献度为依据设计了删除策略。如果对于连续输入的若干个样本, 某隐节点的输出以及输出的连接权值都较小, 则判定

此隐节点已不活跃, 对网络输出的影响较小, 可以删除。具体表示为: 对于每一个学习样本 $[x_t - y (x_t)]$, 按式 (2) 计算每个隐节点的输出 φ_i , 并将其归一化, 即:

$$R_i^N = \frac{\| \varphi_i^N \|}{\| \varphi_{\max}^N \|} < \zeta \tag{11}$$

同时归一化每个隐节点与输出节点的连接权 ω_i , 即:

$$v_i^N = \frac{\| \omega_i^N \|}{\| \omega_{\max}^N \|} < \xi \tag{12}$$

式中 ζ 和 ξ 为设定阈值, N 表示连续的 n 个样本。如果同时满足式 (11)、(12), 则判定该隐节点无效, 应予删除。

1.2.3 网络参数调整策略及分级学习率因子

对于输出层的连接权值, 采用递推最小二乘 (RLS) 算法进行调整。该算法的基本思想为: 每一时刻的迭代是由输入学习样本形成的协方差矩阵递推求得权值的精确解, 即误差梯度为 0 时的解。该算法具有迭代的方向性强、收敛速度快等优点。具体算法推理过程不在本文详述, 得到的权值更新公式如下:

$$\omega (k) = \omega (k-1) + g (k) [f (x_t) - \varphi (x_t) \omega (k-1)] \tag{13}$$

$$g (k) = \frac{P (k-1) \varphi (x_t) [f (x_t) - \varphi (x_t) \omega (k-1)]}{\lambda + \varphi (x_t) P (k-1) \varphi^T (x_t)} \tag{14}$$

$$P (k) = \frac{P (k-1) - g (k) \varphi (x_t) P (k-1)}{\lambda} \tag{15}$$

式中 $g (k)$ 为 κ 时刻的卡尔曼增益, λ 是加权遗忘因子, 它的作用是把过去样本对当前估计值的影响逐渐忘掉, 使所估计的参数尽量反映当前时刻样本的特性。通常的取值范围为 $0 < \lambda < 1$ 。其值越大, 表示过去样本对权值影响越大。 $P (k)$ 为 $R (k)$ 的导数, $R (k)$ 为:

$$R (k) = \sum_{i=1}^k \lambda^{k-i} \varphi (x_i) \varphi (x_i)^T + \lambda R (k-1) + \varphi (x_k) \varphi (x_k)^T \tag{16}$$

由隐层节点参数设置策略可知, 其节点中心即为样本输入值, 这就可能出现该中心距离最优中心存在一定偏差的情况。为了校正偏差, 采用梯度下降法动态调整各节点中心值和输出偏移, 具体公式为:

$$c_i (k+1) = c_i (k) + \Delta c_i \tag{17}$$

$$\alpha(k+1) = \alpha(k) + \Delta\alpha \quad (18)$$

式中 Δc_i 和 $\Delta\alpha$ 分别为当前神经网络函数对 c_i 和 α 的梯度:

$$\Delta c_i = 2 \frac{\eta}{\sigma_i^2(k)} \omega_i(k) \varphi_i(k) (f(x_i) - y(x_i)) (x_i - c_i(k)) \quad (19)$$

$$\Delta\alpha = \eta (y(x_i) - f(x_i)) \quad (20)$$

式中 η 即为学习率因子。传统的 RBF 在线学习算法中, 都是将设定为固定值, 且完全依靠算法设计者的经验而定, 不能根据算法学习过程的需要进行优化调整, 这就导致往往不能得到最佳的网络学习结果。针对这一问题, 本文提出分级学习率因子的概念, 即将学习率因子划分成互不重叠的 n 个不同等级 $\eta_1, \eta_2 \cdots \eta_{n-1}, \eta_n$, n 值可以根据实际应用的需要设置。这里选用样本的理论输出和网络实际输出的误差绝对值作为划分等级的依据, 不同的误差值属于不同的误差区间, 不同的误差区间对应相应等级的学习率因子, 这就实现了在不需要大幅度增加计算开销的情况下, 一定程度上解决了学习率这一参数的自适应问题。具体原理用伪代码表示如下:

IF (误差绝对值 $\in \sigma_1$) 则学习率因子 = η_1 ;

IF (误差绝对值 $\in \sigma_2$) 则学习率因子 = η_2 ;

IF (误差绝对值 $\in \sigma_3$) 则学习率因子 = η_3 ;

...

IF (误差绝对值 $\in \sigma_n$) 则学习率因子 = η_n 。

其中 $\sigma_1, \sigma_2 \cdots \sigma_n$ 为误差绝对值的区间范围。

1.3 改进型 RBF 在线学习算法实现步骤

本文提出的改进型 RBF 在线学习算法具体实现步骤如下: ①初始化网络, 隐层节点数为 0, 设置各项参数; ②对每个在线输入的新样本 $(x_i, y(x_i))$, 分别按式 (2) 和式 (3) 计算各个隐节点的输出及网络的输出; ③根据式 (4)、(5)、(7) 判断是否满足“新颖性”标准, 如满足条件则增加新节点, 由式 (8)、(9)、(10) 设置新节点参数; ④如不满足“新颖性”标准, 则采用递推最小二乘法按式 (13)、(14)、(15) 调整输出权值, 再利用梯度下降法和分级学习率按式 (17) ~ (20) 调节隐层节点中心和输出偏移; ⑤按式 (11)、(12) 判断是否有冗余节点, 如有, 删除; ⑥输入新样本, 转至步骤②继续学习, 直到达到预定学习次数。

2 仿真实验

为了验证改进型 RBF 神经网络在线学习算法

的性能, 本文选用包含有自然指数、乘积、三角函数等多种非线性项的复杂多项式函数进行曲线拟合仿真实验, 并将相关结果同 RAN 算法和文献 [1] 算法进行比较。仿真环境选择 VC++6.0, 编程语言为常用的 C 语言。

曲线拟合多项式为:

$$y = 2e^{-x^2} (x - \cos x + 2x^2) \quad (21)$$

其中 $x \in R^N$, R 为任意实数。相关参数初始化如下: 样本数 $N=400$, 样本输入服从区间 $(-3, 3)$ 内的均匀分布, 重叠系数 $K=0.87$, 最大距离分辨率 $\delta_{\max}=2$, 最小距离分辨率 $\delta_{\min}=0.4$, 误差分辨率 $\epsilon_1=0.02$, $\epsilon_2=0.02$, 冗余节点删除阈值 $\zeta=0.001$, $\xi=0.001$, 分辨率衰减常数 $\tau=0.92$, 最大学习次数 100, 滑动窗口长度 $M=10$, 加权遗忘因子 $\lambda=0.98$, 学习率 η 分为 4 个等级: $\eta_1=0.04$, $\eta_2=0.06$, $\eta_3=0.08$, $\eta_4=0.10$, 对应的误差绝对值区间分别为: $[0, 0.6]$, $[0.6, 0.8]$, $[0.8, 1.0]$, $[1.0, \infty]$ 。学习过程结束后所得结果如表 1 所示, 输出偏移 $\alpha = -0.515303$ 。

表 1 学习结束后得到的网络结果

Table 1 The network results obtained after the study

节点编号	节点中心	连接权值	节点宽度
0	-0.101 596	-1.931 932	0.755 458
1	1.633 866	1.156 786	1.302 799
2	-1.223 854	1.101 265	0.694 217
3	0.738 442	1.338 542	0.523 038
4	1.190 115	0.753 826	0.365 696
5	0.000 000	0.002 335	0.681 071
6	-2.703 779	0.533 744	1.131 955
7	-2.170 740	0.046 985	0.446 725
8	-0.900 796	0.299 795	0.397 967
9	0.000 000	0.002 335	0.322 598
10	-1.752 478	0.061 172	0.314 031
11	0.473 082	0.086 675	0.303 891
12	2.939 548	0.159 373	0.369 192
13	2.091 525	-0.162 610	0.327 745
14	-0.502 210	0.027 243	0.314 476

RBF 神经网络在线学习过程完成后, 程序将随机产生 201 个测试样本用于对网络性能进行测试, 获得测试样本的均方根误差值, 并将结果与 RAN 算法和文献 [1] 算法进行对比, 如表 2、表 3 所示。

表 2 3 种算法隐层节点数对比

Table 2 Comparison of the number of hidden nodes of three algorithms

算法版本	RAN 算法	文献 [1] 算法	本文算法
隐层节点数	10	15	15

由表 2 和表 3 可以看出, 本文提出的改进型在

线学习算法与 RAN 算法均方根误差比较发现：所有测试样本所得到的均方根误差值均大大降低，非线性拟合精度有了显著提高，虽然由 10 个隐层节点增加为 15 个。与文献 [1] 算法相比，隐层节点

数量相同，均包含 15 个隐层节点，但是除了开始的 7 个测试样本的均方根误差值略有升高以外，其它 194 个的均方根误差均有不同程度的降低。

表 3 3 种算法均方根误差对比

Table 3 Comparison of the RMSE of three algorithms

测试样本编号	RAN 算法	文献 [1] 算法	本文算法	绝对误差比较	
				RAN 算法	文献 [1] 算法
0	0.426 130	0.016 476	0.017 752	-0.408 378	+0.001 276
1	0.374 500	0.013 541	0.014 774	-0.359 726	+0.001 233
2	0.328 871	0.011 250	0.012 382	-0.316 489	+0.001 132
3	0.290 365	0.009 794	0.010 730	-0.279 635	+0.000 936
4	0.259 966	0.009 305	0.009 945	-0.250 021	+0.000 640
5	0.238 226	0.009 699	0.010 006	-0.228 220	+0.000 307
6	0.224 903	0.010 693	0.010 712	-0.214 191	+0.000 019
7	0.218 800	0.011 988	0.011 795	-0.207 005	-0.000 193
8	0.217 998	0.013 378	0.013 040	-0.204 958	-0.000 338
9	0.220 372	0.014 741	0.014 307	-0.206 065	-0.000 434
10	0.224 047	0.016 012	0.015 516	-0.208 531	-0.000 496
...					
194	0.121 149	0.067 424	0.066 205	-0.054 944	-0.001 219
195	0.121 809	0.067 684	0.066 471	-0.055 338	-0.001 213
196	0.122 285	0.067 838	0.066 630	-0.055 655	-0.001 208
197	0.122 506	0.067 890	0.066 687	-0.055 819	-0.001 203
198	0.122 457	0.067 854	0.066 654	-0.055 803	-0.001 200
199	0.122 202	0.067 746	0.066 549	-0.055 653	-0.001 197
200	0.121 911	0.067 592	0.066 397	-0.055 514	-0.001 195

3 结 论

本文在研究传统 RBF 在线学习算法的基础上，针对学习过程中学习率固定不变的缺点，引入了分级学习率因子，提出了一种改进型 RBF 在线学习算法，根据理论输出和网络实际输出的误差绝对值选择适当的学习率因子，同时保留了传统在线学习方法自适应调整隐层节点数目、位置、网络权值以及对学习样本没有任何先验知识要求的优点。软件仿真结果表明，该算法误差小精度高，具有良好的应用前景。

参考文献：

[1] NING Guang Liang, ZHANG Xiao Ming. An improved RBF network on - line learning algorithm [C] // 2nd International Symposium on Information Science and Engineering. Shanghai: IEEE Computer Society, 2009: 547 - 552.

[2] 乔俊飞, 韩红桂. RBF 神经网络的结构动态优化设计 [J]. 自动化学报, 2010, 36 (6): 865 - 872.

[3] 梁斌梅, 韦琳娜. 改进的径向基函数神经网络预测模型 [J]. 计算机仿真, 2009, 26 (2): 191 - 194.

[4] 崔海青, 刘希玉. 基于粒子群算法的 RBF 网络参数优化算法 [J]. 计算机技术与发展, 2009, 19 (12): 117 - 119.

[5] 邢桂华, 朱庆保. 基于蚂蚁算法的 RBF 网络参数的两阶段优化 [J]. 计算机工程与应用, 2005 (27): 56 - 59.

[6] 王宏刚, 钱锋. 基于遗传算法的前向神经网络结构优化 [J]. 控制工程, 2007, 14 (4): 387 - 390.

[7] 甘敏, 彭晓燕, 彭辉. RBF 神经网络参数估计的两种混合优化算法 [J]. 控制与决策, 2009, 24 (8): 1172 - 1176.

[8] 张刚林, 刘光灿. 基于一种进化模型的 RBF 网络参数优化 [J]. 控制工程, 2010, 17 (3): 313 - 315.

[9] 卿启新, 叶汉民. 基于模糊 RBF 神经网络的无刷直流电机调速控制 [J]. 化工自动化及仪表, 2010, 37 (7): 84 - 86.

[10] 王君, 朱莉, 蔡之华. 一种基于卡尔曼滤波和模糊控制的 RBF 神经网络新型学习算法 [J]. 计算机应用, 2006, 26 (7): 1700 - 1705.

[11] 张辉, 柴毅. 一种改进的 RBF 神经网络参数优化方法 [J]. 计算机工程与应用, 2012, 48 (20): 146 - 149.

[12] 王剑, 薛飞. 一种 RBF 神经网络的自适应学习算法 [J]. 现代电子技术, 2011, 34 (3): 141 - 143.